

# THE CHARACTERISTIC-BASED-SPLIT PROCEDURE: AN EFFICIENT AND ACCURATE ALGORITHM FOR FLUID PROBLEMS

O.C. ZIENKIEWICZ<sup>a,\*</sup>, P. NITHIARASU<sup>a</sup>, R. CODINA<sup>b</sup>, M. VÁZQUEZ<sup>b</sup> AND P. ORTIZ<sup>c</sup>

<sup>a</sup> *Institute for Numerical Methods in Engineering, Department of Civil Engineering, University of Wales Swansea, Swansea SA2 8PP, UK*

<sup>b</sup> *International Center for Numerical Methods in Engineering, Universidad Politécnica de Catalunya, Barcelona, Spain*

<sup>c</sup> *Centro de Estudios de Técnicas Aplicadas, Cedex Alfonso XII, 3, E-28014, Madrid, Spain*

## SUMMARY

In 1995 the two senior authors of the present paper introduced a new algorithm designed to replace the Taylor–Galerkin (or Lax–Wendroff) methods, used by them so far in the solution of compressible flow problems. The new algorithm was applicable to a wide variety of situations, including fully incompressible flows and shallow water equations, as well as supersonic and hypersonic situations, and has proved to be always at least as accurate as other algorithms currently used. The algorithm is based on the solution of conservation equations of fluid mechanics to avoid any possibility of spurious solutions that may otherwise result. The main aspect of the procedure is to split the equations into two parts, (1) a part that is a set of simple scalar equations of convective–diffusion type for which it is well known that the characteristic Galerkin procedure yields an optimal solution; and (2) the part where the equations are self-adjoint and therefore discretized optimally by the Galerkin procedure. It is possible to solve both the first and second parts of the system explicitly, retaining there the time step limitations of the Taylor–Galerkin procedure. But it is also possible to use semi-implicit processes where in the first part we use a much bigger time step generally governed by the Peclet number of the system while the second part is solved implicitly and is unconditionally stable. It turns out that the characteristic-based-split (CBS) process allows equal interpolation to be used for all system variables without difficulties when the incompressible or nearly incompressible stage is reached. It is hoped that the paper will help to make the algorithm more widely available and understood by the profession and that its advantages can be widely realised. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: CBS algorithm; fluid problems; convective–diffusion-type

## 1. INTRODUCTION

In recent years, the current authors have published several papers concerning the basis and applications of a universal algorithm for practically all fluid mechanics problems, but no specific name has been given to it [1–7]. From now on, it shall be referred to as the

---

\* Correspondence to: Institute for Numerical Methods in Engineering, Department of Civil Engineering, University of Wales Swansea, Swansea SA2 8PP, UK.

characteristic-based-split (CBS) algorithm, which explains the main motivation for its origin. The present paper summarises the theory of the algorithm as well as demonstrates its practical applications.

The algorithm can be used in explicit, semi-implicit, nearly implicit and, indeed, in fully implicit forms. Its advantages can be fully illustrated, however, using explicit and semi-implicit forms and this paper will concentrate on these. The algorithm was initially based on a series of preliminary studies conducted between 1990 and 1995 [8–12], but the split was introduced correctly only later, in Reference [1].

Before the present algorithm was available, most successes of finite difference and finite element methods in fluid mechanics were based on some variant of the Lax–Wendroff [13] scheme, which by approximating better the time derivative also introduced a stabilisation of the convective terms. The method that applied to finite elements become known as the Taylor–Galerkin method [14,15]. Later it was discovered that for scalar variables, a direct algorithm utilising the optimal approximation along the characteristics, the characteristic Galerkin method, could be shown to be identical to the Taylor–Galerkin method [16,17].

The only similarity between the two procedures exists in the scalar case described by the well-known convection–diffusion equation that is often used as a model for fluid mechanics problems. However, for problems involving several variables, typical of fluid mechanics, the application of the characteristic Galerkin method is not possible as only a single characteristic speed must be involved. For this reason, the Taylor–Galerkin procedure has been used widely even though giving a suboptimal approximation. Motivation for the development of the present algorithm came from the fact that the characteristic Galerkin procedure for a scalar variable is optimal in the sense of approximation and that, with suitable splitting, it can be applied for the first stage of the solution of the fluid mechanics equation. The reminder of the split being self-adjoint can then be solved optimally using the Galerkin procedure.

This split follows the general process initially introduced by Chorin [18,19] for incompressible flow problems. Further motivation, which originated in the new algorithm, is based on an additional benefit coming from the fact that, as already observed, for incompressible situations, the algorithm permitted equal (and indeed arbitrary) interpolations for all variables to be used. This sidesteps the Babuska–Brezzi requirement in finite elements, and similarly all difficulties in standard finite difference schemes. In the present form, the solution turns out to be fully accurate with arbitrary interpolation for velocity and pressure for full incompressibility.

This indeed widens the spectrum of solutions using the new algorithm and indeed it is shown later how, in dealing with problems of solid dynamics, the present algorithm proves to be capable of generalising such explicit approaches as those given by DYNA [20,21].

Although the original Chorin split [18,19] could never be used in a fully explicit code without introducing artificial or real compressibility, the split provides a fully explicit algorithm that is valid at least for steady state problems, even in a fully incompressible range. When compressibility exists, computational advantages of the explicit form compare well with other currently used schemes and the additional cost due to splitting the operator is insignificant. Generally, for an identical cost, the results are considerably improved even in standard range of aerodynamical problems. However, a further advantage is that both subsonic and supersonic problems can be solved by the same code.

The algorithm involves time integration and, in general, the time step size will be limited by the nature of the time stepping procedure adopted in each part of the split. If a fully explicit procedure is adopted in the first part of the split then the time step is governed by a Courant number defined in terms of the flow velocity  $|\mathbf{u}|$  and viscosity  $\nu$ . If an explicit scheme is also

adopted for the second part of the split, then a Courant number depending on the compressible wave celerity  $c$  is invoked and may control the time step size. We shall thus have several possible categories of problems.

(a) *Transonic and supersonic flows, fully explicit process*

Here, in general, fully explicit computations are preferred as the time step limitation for both parts of the split are similar.

(b) *Low Mach number with low viscosity, semi-implicit process*

Here, the flow is nearly incompressible and the time step used will be governed by first part of the split, with the Courant number not affected much by viscosity.

(c) *Low Mach number with high viscosity, nearly implicit process*

In this range of flow, the limiting time step again is governed by the first part of the split, but here viscosity may pose very severe limits. In such a regime, a 'nearly' implicit procedure is recommended in which the viscous terms are treated implicitly.

(d) *Shallow water equations, semi-implicit or nearly implicit*

Here, fully explicit computations have been efficiently used by Peraire *et al.* [22]. However, semi-implicit treatment can increase the time step in tidal computations by one (or even two) orders of magnitude [23–25]. In such problems, if horizontal viscosity is large, a 'nearly' implicit process may be considered.

Typical examples for each of the categories will be shown later. Indeed, the procedure used in (e) *fully explicit dynamic computation in solids* will be demonstrated. However, many further applications of the CBS algorithm are possible and will be reported below.

## 2. THE SCALAR CONVECTION–DIFFUSION PROBLEM AND THE CHARACTERISTIC GALERKIN EXPLICIT APPROXIMATION

Before proceeding with the description of the full algorithm, we shall recall the application of the characteristic Galerkin method in the explicit form to a typical convection–diffusion process with a scalar dependent variable  $\phi$ .

$$\frac{\partial V}{\partial t} + \frac{\partial F_i}{\partial x_i} + \frac{\partial G_i}{\partial x_i} + Q = 0, \quad (1)$$

where  $x_i$  is the  $i$ th co-ordinate ( $i = 1, 2, 3$ ),

$$F_i = u_i \phi \quad (2)$$

is the convected flux,

$$G_i = -k \frac{\partial \phi}{\partial x_i} \quad (3)$$

is the diffusion flux,

$$Q = Q(x, t) \quad (4)$$

is the source term, and

$$\mathbf{u} = \mathbf{u}(\mathbf{x}, t) \quad (5)$$

is the velocity field that is assumed to be known.

The full equation can thus be alternatively written as

$$\frac{\partial \phi}{\partial t} = -u_j \frac{\partial \phi}{\partial x_j} + \frac{\partial}{\partial x_i} \left( k \frac{\partial \phi}{\partial x_i} \right) - Q - \phi \frac{\partial u_j}{\partial x_j} = R(\phi), \quad (6)$$

in which only the first term on the right-hand-side is not self-adjoint. As that term corresponds precisely to an advection wave moving with a velocity  $\mathbf{u}$ , a change of co-ordinates to the characteristic ones, given by

$$dx'_i = dx_i - u_i dt, \quad (7)$$

makes the offending term vanish leaving a fully self-adjoint system.

For such a self-adjoint system it is known that the standard Galerkin approximation in space is optimal, but the inconvenience of a moving co-ordinate system is introduced. However, this can be overcome with suitable remeshing and the procedure has been used in many early solutions of the above equation. Here, Adye–Brrebia (1974) [26] appears to be first followed by much later work [27–31]. While the exact co-ordinate transformation introduced no error, simplified procedures using a Taylor approximation within the time step eliminated the costly process of remeshing (or mesh interpolation), introducing, however, a time step limitation [32]. This process is fully described in References [1,2] and the explicit form can be written in fixed co-ordinates as

$$\Delta \phi = \phi^{n+1} - \phi^n = \Delta t R(\phi)^{n+\theta_3} - \frac{\Delta t^2}{2} u_i \frac{\partial}{\partial x_i} (R(\phi))^n + O(\Delta t^3), \quad (8)$$

where  $0 \leq \theta_3 \leq 1$  and  $R(\phi)$  is defined by Equation (6).

This, as already mentioned, is of an identical form to that resulting from the Taylor–Galerkin procedure and the second term adds to the stabilizing diffusion in the streamline direction. Indeed, a similar form can be obtained here in a variety of ways, and recently Oñate [33] introduced an interesting form of stability control by using so-called ‘finite increment’ equations. However, only the characteristic form ensures the optimal approximation, as Equation (8) is derived from a self-adjoint form in which spatial discretization by the Galerkin method can be optimally used. We can thus write the approximation

$$\phi = N \bar{\phi} \quad (9)$$

and use the weighting  $N^T$  in the integrated residual expression. Thus, we obtain

$$\mathbf{M}(\bar{\phi}^{n+1} - \bar{\phi}^n) = \Delta t[(\mathbf{C}\bar{\phi}^n + \mathbf{K}\bar{\phi}^n + \mathbf{f}^n) - \Delta t(\mathbf{K}_u\bar{\phi}^n + \mathbf{f}_s^n)], \quad (10)$$

if we assume  $\theta_3 = 0$  as is generally done in an explicit form and omit higher derivatives and source terms. In the above equation,

$$\begin{aligned} \mathbf{M} &= \int_{\Omega} N^T N \, d\Omega, & \mathbf{C} &= \int_{\Omega} N^T \frac{\partial}{\partial x_i} (u_i N) \, d\Omega, \\ \mathbf{K} &= \int_{\Omega} \frac{\partial N^T}{\partial x_i} k \frac{\partial N}{\partial x_i} \, d\Omega, & \mathbf{f} &= \int_{\Omega} N^T Q \, d\Omega + \text{b.t.} \end{aligned} \quad (11)$$

and  $\mathbf{K}_u$  and  $\mathbf{f}_s^n$  come from the new term introduced by the discretization along the characteristics. After integration by parts, the expression of  $\mathbf{K}_u$  and  $\mathbf{f}_s$  is

$$\begin{aligned} \mathbf{K}_u &= -\frac{1}{2} \int_{\Omega} \frac{\partial}{\partial x_i} (u_i \mathbf{N}^T) \frac{\partial}{\partial x_i} (u_i \mathbf{N}) \, d\Omega, \\ \mathbf{f}_s &= -\frac{1}{2} \int_{\Omega} \frac{\partial}{\partial x_i} (u_i \mathbf{N}^T) \mathcal{Q} \, d\Omega + \text{b.t.}, \end{aligned} \quad (12)$$

where b.t denotes integrals along region boundaries [32].

The approximation is valid for any scalar convected quantity even if it is the velocity  $u_i$  component itself, as is the case with the momentum conservation equations. For this reason we have elaborated the full details of the spatial approximation above as the matrices will be repeatedly used.

It is of interest that the explicit form of Equation (10) is only conditionally stable. For one-dimensional problems, the stability condition is given as (neglecting the effect of sources)

$$\Delta t \leq \Delta t_{\text{crit}} = \frac{h}{|\mathbf{u}|} \left( \sqrt{\frac{1}{Pe^2} + \frac{1}{3}} - \frac{1}{Pe} \right) \quad (13)$$

for linear elements. In this equation, the Peclet number  $Pe$  is defined as

$$Pe = \frac{|\mathbf{u}|h}{2k}. \quad (14)$$

In two-dimensional problems, the critical time step may be computed as [34]

$$\Delta t_{\text{crit}} = \frac{\Delta t_{\sigma} \Delta t_v}{\Delta t_{\sigma} + \Delta t_v}, \quad (15)$$

where  $\Delta t_{\sigma}$  is given by Equation (13) and  $\Delta t_v = h^2/2k$  is the diffusive limit for the critical one-dimensional time step.

Further, with  $\Delta t = \Delta t_{\text{crit}}$ , the steady state solution results in a diffusion change almost identical to that obtained using the optimal streamline upwinding procedures [32]. Thus, if steady state solutions are the main objective of the computation, such a value of  $\Delta t$  should be used in connection with the  $\mathbf{K}_u$  term.

In Equation (8) we have introduced a  $\theta_3$  parameter which, when  $> \frac{1}{2}$ , stabilizes the explicit solution. A fully implicit form is, however, expensive, involving non-linear, unsymmetric matrices. However, it is often convenient to apply  $\theta_3 \geq \frac{1}{2}$  to the diffusive term only. We call this a *nearly implicit* form and if employed we return to the stability condition

$$\Delta t_{\text{crit}} = \frac{h}{|\mathbf{u}|}, \quad (16)$$

which can present an appreciable benefit.

### 3. THE GENERAL FRACTIONAL STEP ALGORITHM FOR THE NAVIER-STOKES EQUATIONS

#### 3.1. The equations of flow—Navier–Stokes problem

The full conservation form of the Navier–Stokes equations for compressible flow is traditionally written as

$$\frac{\partial V}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} + \frac{\partial \mathbf{G}_i}{\partial x_i} + \mathcal{Q} = 0, \quad (17)$$

with

$$\mathbf{V}^T = (\rho, \rho u_1, \rho u_2, \rho E) \quad (18)$$

being the independent variable vector,

$$\mathbf{F}_i^T = (\rho u_i, \rho u_i u_1, \delta_{i1} p, \rho u_i u_2 + \delta_{i2} p, u_i(\rho E + p)) \quad (19)$$

defining the convective flux vector, and

$$\mathbf{G}_i^T = (0, -\tau_{i1}, -\tau_{i2}, q_i - \tau_{ij} u_j) \quad (20)$$

defining the diffusion fluxes. Finally,

$$\mathbf{Q}^T = (0, \rho g_1, \rho g_2, \rho(g_i u_i + q''')) \quad (21)$$

gives the source terms.

In the above, the stress components  $\tau_{ij}$  are related to velocity gradients by

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (22)$$

where  $u_i$  are the velocity components;  $\rho$  is the density;  $E$  is the specific energy;  $q$  is the heat flux;  $q'''$  is the heat generation per unit mass;  $p$  is the pressure; and  $g_i$  is the acceleration due to gravity.

The equations are completed by the universal gas law

$$p = \rho RT, \quad (23)$$

where  $R$  is the gas constant.

The sound velocity is defined assuming constant entropy as

$$c^2 = \frac{\partial p}{\partial \rho} = \frac{\gamma p}{\rho}, \quad (24)$$

where  $\gamma$  is the ratio of specific heats.

Further, we can write conveniently

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial p} \frac{\partial p}{\partial t} = \frac{1}{c^2} \frac{\partial p}{\partial t}, \quad (25)$$

though this expression assumes again constant entropy and is therefore only an approximation. In the following, we shall use Equation (25) but elsewhere we discuss the possibility of correcting any errors involved by amendment of the algorithm [7].

While in the gas flow all the equations are fully coupled, for incompressible flows in which  $c = \infty$ , the energy equations can be solved independently after the velocity field has been established. Nevertheless, a single algorithm for the solution of both problems is possible as we shall now show.

Although the form of Equation (17) is identical to that of the convection–diffusion problem of Equation (1), three wave speeds exist and the characteristic Galerkin procedure can not be directly applied. In the next section we show how this can be employed with a fractional step.

In Appendix A, we show the depth averaged, shallow water equations. The format of these is precisely the same as that of the Navier–Stokes equations and all of the procedures discussed above can be again used for that class of problems.

### 3.2. Characteristic-based-split algorithm

For convenience we shall rewrite Equation (17) in a more direct form, omitting initially the energy equation. These equations can be solved completely in a time increment  $\Delta t$  as the only coupling that exists is through the speed of sound  $c$  for which we shall simply use the value at time  $t_n$  due to the explicit nature of the time stepping algorithm.

We thus write the first form of Equation (17), i.e. the mass flow continuity, as

$$\frac{\partial \rho}{\partial t} = \frac{1}{c^2} \frac{\partial p}{\partial t} = - \frac{\partial U_i}{\partial x_i}, \quad (26)$$

in which we use Equation (25). Similarly, for each of the momentum conservation equations, we write

$$\frac{\partial U_i}{\partial t} = - \frac{\partial}{\partial x_j} (u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial p}{\partial x_i} - g_i. \quad (27)$$

In the above we define the mass flow fluxes as

$$U_i = \rho u_i \quad (28)$$

and note that we can discretize Equation (27) in time using the characteristic Galerkin process as, except for the pressure term, this equation is similar to the convection–diffusion *viz.* Equation (8). Therefore, with the variable being  $U_i$ , we have

$$U_i^{n+1} - U_i^n = \Delta t \left[ \frac{\partial}{\partial x_j} (u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left( \frac{\partial}{\partial x_j} (u_j U_i) + g_i \right) \right]^n - (1 - \theta_2) \Delta t \left[ \frac{\partial p}{\partial x_i} - \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left( \frac{\partial p}{\partial x_i} \right) \right]^n - \theta_2 \Delta t \left[ \frac{\partial p}{\partial x_i} \right]^{n+1}. \quad (29)$$

It is observed that the pressure term has now been evaluated at  $t^n + \theta_2 \Delta t$  and is not treated explicitly. Before proceeding further it is convenient to introduce an auxiliary variable  $U_i^*$  such that

$$\Delta U_i^* = U_i^{*n+1} - U_i^n = \Delta t \left[ - \frac{\partial}{\partial x_j} (u_j U_i) + \frac{\partial \tau_{ij}}{\partial x_j} - g_i + \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \left( \frac{\partial}{\partial x_j} (u_j U_i) + g_i \right) \right]^n \quad (30)$$

represents the first explicit part of the split and the correction given below is available once the pressure increment is evaluated.

$$\begin{aligned} \Delta U_i &= U_i^{n+1} - U_i^n = \Delta U_i^* - \Delta t (1 - \theta_2) \left[ \frac{\partial p}{\partial x_i} - \frac{\Delta t}{2} u_k \frac{\partial}{\partial x_k} \frac{\partial p}{\partial x_i} \right]^n - \theta_2 \Delta t \left[ \frac{\partial p}{\partial x_i} \right]^{n+1} \\ &= \Delta U_i^* - \Delta t \left[ \frac{\partial p^n}{\partial x_i} + \theta_2 \frac{\partial \Delta p}{\partial x_i} \right] + (1 - \theta_2) \frac{\Delta t^2}{2} u_k \frac{\partial}{\partial x_k} \frac{\partial p^n}{\partial x_i}. \end{aligned} \quad (31)$$

In the above,  $\Delta p = p^{n+1} - p^n$  and the last term, as before, represents the ‘source’ correction. From Equation (26) we have, omitting third-order terms

$$\Delta p = \left( \frac{1}{c^2} \right)^n \Delta p = - \Delta t \frac{\partial U_i^{n+\theta_1}}{\partial x_i} = - \Delta t \left[ \frac{\partial U_i^n}{\partial x_i} + \theta_1 \frac{\partial \Delta U_i^*}{\partial x_i} - \Delta t \theta_1 \left( \frac{\partial^2 p^n}{\partial x_i \partial x_i} + \theta_2 \frac{\partial^2 \Delta p}{\partial x_i \partial x_i} \right) \right]. \quad (32)$$

It is clear that the equations can be solved after spatial discretization in the following order:

Equation (30) to obtain  $\Delta U_i^*$ ,

Equation (32) to obtain  $\Delta p$ ,

Equation (31) to obtain  $\Delta U_i$ , thus establishing the values at  $t^{n+1}$ .

It is important to remark that this sequence allows one to solve in an efficient manner and with the adequate numerical damping of the continuous equations (26) and (27), i.e. the conservation form of the Navier–Stokes equations. Therefore, this algorithm is well suited for dealing with supersonic and hypersonic problems in which the conservation form ensures that shocks will be placed at right position.

In all of the equations given below the standard Galerkin procedure is used with a discretization

$$U_i = N\bar{U}_i, \quad \Delta U_i = N\Delta\bar{U}_i, \quad \Delta U_i^* = N\Delta\bar{U}_i^* \quad (33)$$

and

$$p = N_p\bar{p}. \quad (34)$$

This gives from Equation (30) the solution for  $\bar{U}_i^*$  as

### Step 1

$$\Delta\bar{U}_i^* = -M^{-1}\Delta t[(C\bar{U} + K\bar{U} - f) - \Delta t(K_u\bar{U} + f)]^n, \quad (35)$$

where all the discretization matrices are the same as those defined by Equations (11) and (12).

Similarly, the discretization of Equation (32) gives

### Step 2

$$(\tilde{M} + \Delta t^2\theta_1\theta_2H)\Delta\bar{p} = \Delta t[Q(\bar{U}^n + \theta_1\Delta\bar{U}^*) - \Delta t\theta_1H\bar{p} - f_p]^n, \quad (36)$$

which can be solved for  $\Delta\bar{p}$ .

The new matrices arising here are

$$\begin{aligned} H &= \int_{\Omega} \frac{\partial N_p^T}{\partial x_i} \frac{\partial N_p}{\partial x_i} d\Omega, \\ \tilde{M} &= \int_{\Omega} N_p^T \left( \frac{1}{c^2} \right)^n N_p d\Omega, \\ Q &= \int_{\Omega} \frac{\partial N_p^T}{\partial x_i} N d\Omega. \end{aligned} \quad (37)$$

The question of establishing the boundary conditions for the pressure is discussed in detail in Appendix B.

The final stage of the computation of the mass flow vector  $U_i^{n+1}$  is completed by the discretization of Equation (31) and we have now simply

### Step 3

$$\Delta\bar{U} = \Delta\bar{U}^* - M^{-1}\Delta t \left[ Q^T(\bar{p}^n + \theta_2\Delta\bar{p}) + \frac{\Delta t}{2} P\bar{p}^n \right], \quad (38)$$

where



$$\mathbf{P} = (1 - \theta_2) \int_{\Omega} \frac{\partial}{\partial x_i} (u_i \mathbf{N}_p^T) \frac{\partial \mathbf{N}_p^T}{\partial x_i} d\Omega. \quad (39)$$

At the completion of this stage, the values of  $\bar{\mathbf{U}}^{n+1}$  and  $\bar{\mathbf{p}}^{n+1}$  are fully determined but the computation of energy  $\rho E^{n+1}$  is needed so that new values of  $c^{n+1}$ , the speed of sound, can be determined.

The last of Equation (17), i.e. the energy conservation equation, can be written as

$$\frac{\partial(\rho E)}{\partial t} = -\frac{\partial}{\partial x_j} (u_j \rho E) + \frac{\partial}{\partial x_i} \left( k \frac{\partial T}{\partial x_i} \right) - \frac{\partial}{\partial x_j} (u_j p) + \frac{\partial}{\partial x_i} (\tau_{ij} u_j). \quad (40)$$

Once again the equation is identical in form to that of the scalar problem of convection–diffusion (Equation (6)) if we observe that  $p$ ,  $U_p$ , etc., have been determined. Now the last term can be evaluated at time  $(n + \theta_3)$  for improved accuracy but in what follows we shall take  $\theta_3 = 0$  for simplicity.

Using the characteristic Galerkin approximation of Equation (8) and discretizing as

$$\rho E = N_E \bar{E}, \quad (41)$$

we have

#### Step 4

$$\Delta \bar{E} = -\Delta t [C \bar{E}^n + K T^n + f_e^n - \Delta t (K_u \bar{E}^n + f_{es}^n)], \quad (42)$$

where  $\bar{E}$  contains the nodal values of  $\rho E$  and again the matrices are identical to those previously obtained (assuming that  $\rho E$  and  $T$  can be suitably scaled in the conduction term).

Again, the forcing vectors can be appropriately defined as

$$f_e = \int_{\Omega} N_E^T \frac{\partial}{\partial x_j} (u_j p \tau_{ij} u_i) d\Omega + \text{b.t.},$$

$$f_{es} = -\frac{1}{2} \int_{\Omega} \frac{\partial}{\partial x_i} (u_i N_E^T) \frac{\partial}{\partial x_j} (u_j p \tau_{ij} u_i) d\Omega. \quad (43)$$

It is of interest to observe that the process of Step 4 can be extended to include, in an identical way, the equations describing the transport of such quantities as turbulence parameters, chemical concentrations, etc., once the first essential Steps 1–3 have been completed.

At this point it is clear why we call our scheme ‘characteristic based split’. Observe that Equation (29) has been discretized along the characteristic of the total derivative applied to the momentum components  $U_p$ , which is not a characteristic of the whole problem due to the presence of the pressure. However, this pressure does not appear in Equation (30) for the intermediate momentum. Therefore, the characteristic of the total time derivative applied to the momentum can in fact be thought of as the characteristic of the problem for the intermediate momentum. It is in this sense that we call our scheme ‘characteristic based’.

## 4. SEMI-IMPLICIT AND EXPLICIT FORMS OF THE ALGORITHM

The algorithm described can be used in a semi-implicit form and indeed only in this form can incompressible problems in which  $c = \infty$  and  $\bar{\mathbf{M}} = 0$  be solved. Taking

$$\frac{1}{2} \leq \theta_1 \leq 1,$$

$$\frac{1}{2} \leq \theta_2 \leq 1, \quad (44)$$

the algorithm is conditionally stable. The permissible time step is governed by the critical step of the explicit relation solved in Step 1 of the algorithm. This is the standard convection–diffusion problem discussed in Section 2 and the same stability limits apply, reaching a value close to

$$\Delta t_{\text{crit}} = \frac{h}{|\mathbf{u}|} \quad (45)$$

for an inviscid fluid.

For slightly compressible or incompressible problems in which  $\tilde{M}$  is small or zero, the semi-implicit form is efficient and it should be noted that the matrix  $\mathbf{H}$  of Equation (36) does not vary during the computation process and can be partially inverted.

In other semi-implicit forms, when compressibility exists, the question of the correctness of the approximation of Equation (25) can be questionable, although an iterative correction can always be applied.

However, it is possible to revert to a fully explicit form by putting  $\theta_2 = 0$  but still keeping  $\theta_1$  in the interval presented in Equation (44). Now of course the critical step will be reduced to the order of

$$\frac{h}{c + |\mathbf{u}|} \quad (46)$$

and this is indeed the same limit as that encountered in other explicit forms of Euler or Navier–Stokes computational schemes currently used.

Further, Equations (35), (36), (38) and (42) can be solved simultaneously if the correction term on the right-hand side of Equation (42) is omitted. This is an additional approximation and is not necessary but is introduced here to mimic artificial diffusions previously used with the standard Galerkin form.

The use of the approximation of Equation (25) is not necessary in any fully explicit scheme expected as the density increment is directly obtained if we note that

$$\tilde{\mathbf{M}} \Delta \bar{\rho} = \mathbf{M} \Delta \bar{\rho}, \quad (47)$$

With the above simplifications we can return to the original equation (17) and using the Galerkin approximation on this we can write directly

$$\Delta \bar{\mathbf{V}} = -\mathbf{M}^{-1} \Delta t \left[ \int_{\Omega} \mathbf{N}^T \left( \frac{\partial \mathbf{F}_i}{\partial x_i} + \frac{\partial \mathbf{G}_i}{\partial x_i} \right) d\Omega - \frac{1}{2} \Delta t \int_{\Omega} \mathbf{N}^T \mathbf{D} d\Omega \right]^n \quad (48)$$

omitting the source terms for clarity. The added diffusion terms  $\mathbf{D}$  are defined below and have to be integrated by parts in the usual manner.

$$\left\{ \begin{array}{l} 2 \frac{\partial^2}{\partial x_i \partial x_i} p \\ u_i \frac{\partial}{\partial x_i} \left[ \frac{\partial}{\partial x_i} (u_j \rho u_1) + \frac{\partial p}{\partial x_1} \right] \\ u_i \frac{\partial}{\partial x_i} \left[ \frac{\partial}{\partial x_i} (u_j \rho u_2) + \frac{\partial p}{\partial x_2} \right] \\ u_i \frac{\partial}{\partial x_i} \left[ \frac{\partial}{\partial x_i} (u_j \rho u_3) + \frac{\partial p}{\partial x_3} \right] \\ u_i \frac{\partial}{\partial x_i} \left[ \frac{\partial}{\partial x_i} (u_j \rho E + u_j p) \right] \end{array} \right\} . \quad (49)$$

The ‘diffusions’ added are simple and are streamline-oriented (for divergence-free velocities), thus not masking the true effects of viscosity as happens in some schemes.

If only steady state results only are sought it would appear that  $\Delta t$  in the definition of the matrix  $\mathbf{D}$  should be set at its optimal value of  $\Delta t_{\text{crit}} \approx h/|\mathbf{u}|$  and we generally recommend this value for the full scheme.

However, the oversimplified scheme of Equation (49) loses some accuracy and, even when steady state is reached, will give slightly different results to those obtained using the full sequential updating. The small additional cost involved in computing the sequence  $\Delta \bar{\mathbf{U}}^* \rightarrow \Delta \mathbf{p} \rightarrow \Delta \bar{\mathbf{U}} \rightarrow \Delta \bar{\mathbf{E}}$  will have to be balanced against the accuracy increase. In general, we have found that the two-step version is preferable.

It is of interest to note here in passing that the full sequential scheme introduces a so-called ‘forth-order’ diffusion proportional to  $\Delta t(\mathbf{Q}^T \mathbf{M}^{-1} \mathbf{Q} - \mathbf{H})\bar{\mathbf{p}}$  in addition to the second-order diffusion proportional to  $\Delta t \mathbf{H}\bar{\mathbf{p}}$  into the computation. We shall indicate how this arises and how keeping it is useful in the next section.

## 5. THE UNEXPECTED BONUS OF ‘CIRCUMVENTING’ THE BB RESTRICTIONS

We examine here the structure of equations reached in steady conditions. For simplicity we shall consider only the Stokes form of governing equations in which the convective terms disappear. Further, we shall take the fluid as incompressible and thus uncoupled from the energy equations. Now, the three steps of Equations (35), (36) and (38) are written as

$$\begin{aligned} \Delta \bar{\mathbf{U}}^* &= -\Delta t \mathbf{M}^{-1} [\mathbf{K} \mathbf{U}^n - \mathbf{f}], \\ \Delta \bar{\mathbf{p}} &= \frac{1}{\Delta t \theta_1 \theta_2} \mathbf{H}^{-1} [\mathbf{Q}(\bar{\mathbf{U}}^n + \theta_1 \Delta \bar{\mathbf{U}}^*) - \Delta t \theta_1 \mathbf{H} \bar{\mathbf{p}}^n - \mathbf{f}_p], \\ \Delta \bar{\mathbf{U}} &= \Delta \bar{\mathbf{U}}^* - \Delta t \mathbf{M}^{-1} \mathbf{Q}^T (\bar{\mathbf{p}}_n + \theta_2 \Delta \bar{\mathbf{p}}). \end{aligned} \quad (50)$$

In the steady state we have  $\Delta \bar{\mathbf{p}} = \Delta \bar{\mathbf{U}} = 0$  and eliminating  $\Delta \bar{\mathbf{U}}^*$  we can write (dropping the superscript  $n$ )

$$\mathbf{K} \bar{\mathbf{U}} + \mathbf{Q}^T \bar{\mathbf{p}} = \mathbf{f} \quad (51)$$

from the first and third equations of (50) and

$$\mathbf{Q}\bar{\mathbf{U}} + \theta_1 \Delta t \mathbf{Q} \mathbf{M}^{-1} \mathbf{Q}^T \mathbf{p} - \Delta t \theta_1 \mathbf{H} \mathbf{p} - \mathbf{f}_p = 0 \quad (52)$$

from the second and third.

We finally have a system that can be written in the form of

$$\begin{bmatrix} \mathbf{K} & \mathbf{Q}^T \\ -\mathbf{Q} & \Delta t \theta_1 [\mathbf{H} - \mathbf{Q} \mathbf{M}^{-1} \mathbf{Q}^T] \end{bmatrix} \begin{Bmatrix} \bar{\mathbf{U}} \\ \bar{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix}, \quad (53)$$

where  $\mathbf{f}_1$  and  $\mathbf{f}_2$  arise from the forcing terms.

This system has a non-zero diagonal that is proportional to  $\Delta t$  and which, as already mentioned, is very similar to the forms suggested by other reasoning [32] to avoid matrix singularity and ensure stability. Further, the system is always positive definite.

Also, it should be noticed that if the additional simplification introduced into Equations (48) and (49) is made to avoid the sequential operations, the diagonal term becomes  $\Delta t \theta_1 \mathbf{H}$  and we still avoid BB conditions.

However, it can be easily verified that if the pressure gradient term is retained in Equation (30) (which would seem to give a better approximation), the diagonal term of Equation (53) is identically zero and the BB conditions in the full scheme can not be avoided.

## 6. SOME SOLUTION OF TYPICAL EXAMPLES

In this section we illustrate the applications and show the advantages gained by the use of the CBS algorithm in various classes of problems.

We shall follow the classification of problems into various categories stated in the introduction, but some general remarks will be made as these arise. In all of the problems discussed in this paper, the same computer coding was used and only the simplest element, i.e. the linear triangular, is used in all examples.

The application of higher-order elements and elements of different nature is of course applicable under certain circumstances but we shall not make detail comments on those here.

### 6.1. Fully explicit procedure, subsonic, transonic and supersonic flows

The use of a fully explicit procedure is preferred in aeronautical computations and the CBS algorithm is well suited for this.

We have used in all of the present examples the simplex, linear, triangular elements in two-dimensional form or simple tetrahedral elements in three-dimensional form. However, the use of quadratic (or higher-order) elements is possible, especially when steady state conditions are only sought and when the 'lumped' approximation of the mass matrix causes no significant errors. Examples of the use of such elements are, however, not numerous, with some in [3].

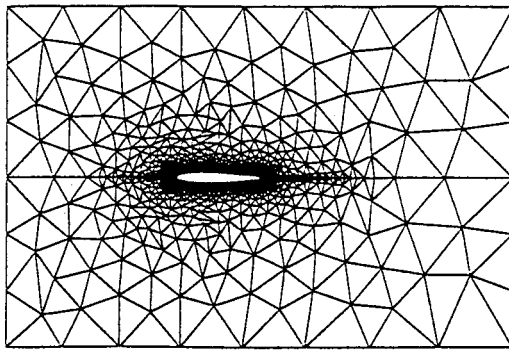
#### Example 1

*NACA 0012 aerofoil with zero angle of attack,  $M = 0.5, 1.2$ .*

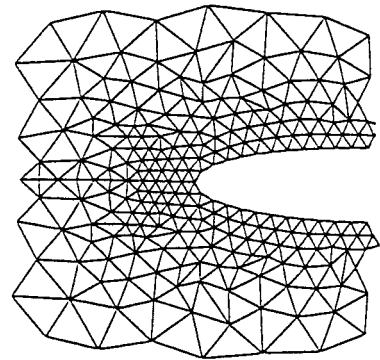
In this example we assess the performance of the CBS in a fully explicit form in subsonic and supersonic situations.

In Figure 1(a) and (b), a typical graded mesh for the study of the NACA 0012 aerofoil is shown. In Figure 1(c)–(e), the results of two Mach numbers (0.5 and 1.2) with the fully explicit form of the multistep and single step forms of the algorithm are presented.

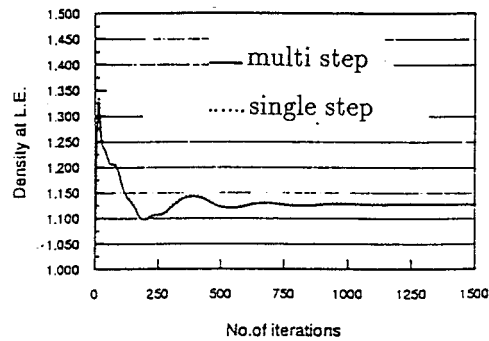
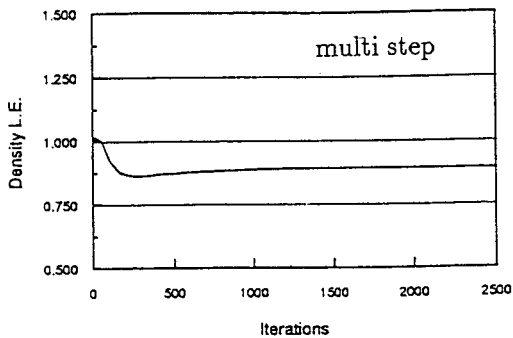
It is of interest to note that for the lower (subsonic) Mach number, almost no difference in performance is observed, while in the supersonic range, the single step algorithm gives some



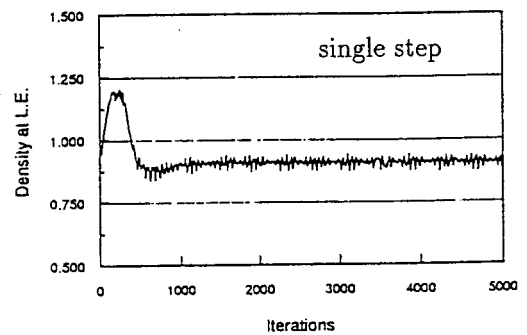
(a)



(b)

(c)  $M = 0.5$ 

(d)



(e)

 $M = 1.2$ 

Figure 1. Inviscid flow past NACA 0012 aerofoil,  $M=0.5$ ,  $\alpha=0^\circ$ . (a) Unstructured mesh of 1824 elements and 969 nodes; (b) details of mesh near leading-edge; (c) convergence for  $M=0.5$  with multi- and single step, fully explicit form; (d) convergence for  $M=1.2$  for multi step scheme; (e) convergence for  $M=1.2$  for single step scheme.

overshoots and oscillations. The conclusion, confirmed by other solutions, is that it is always safe and more accurate to perform the full algorithm procedure.

Figure 2 compares the performance of the new algorithm with the performance of the previously used Taylor–Galerkin scheme.

It is noticed that the CBS algorithm improves the results dramatically near the stagnation solution point without the use of any additional artificial diffusion (which is essential to get any reasonable result using the Taylor–Galerkin scheme).

Indeed, it is rather surprising that even the single step procedure gives nearly the correct pressures/densities at the stagnation without the addition of further stabilizations.

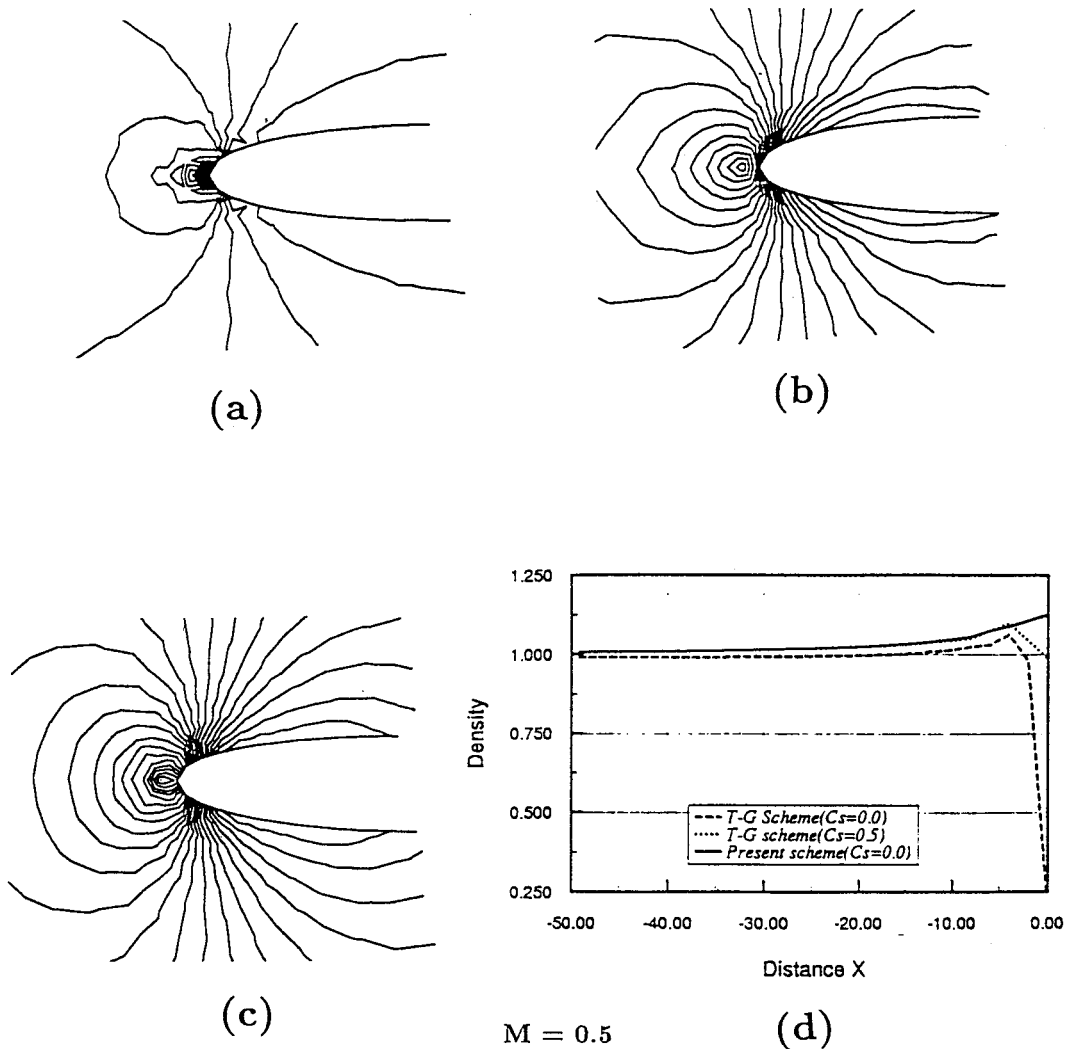


Figure 2. Subsonic inviscid flow past NACA 0012 aerofoil with  $\alpha = 0^\circ$  and  $M = 0.5$ . (a) Density contours with T–G scheme with  $C_s = 0$  (no additional viscosity); (b) density contours with T–G scheme with  $C_s = 0.5$  (with additional viscosity); (c) density contours with CBS scheme with  $C_s = 0$  (no additional viscosity); (d) comparison of density along the stagnation line.

**Example 2**

*Inviscid transonic flow over NACA 0012 aerofoil with 1.2° angle of attack,  $M = 0.85$ .*

In this example a total lift is generated and in addition some shock developments occur. With triangular elements and on an unstructured mesh some adaptivity is used to remesh the problem, as in Figure 3(a) and (b), where the adapted mesh as well as the pressure contours are shown.

In Figure 3(c), the pressure distribution resulting in a lifting force is given and the results are compared with those obtained in an AGARD workshop [35].

**Examples 3 and 4**

*Supersonic and hypersonic inviscid flow past cylinder,  $M = 3, 10$ .*

These two examples are designed to plot the performance of the CBS algorithm in supersonic (or hypersonic) computation for which often very specialized schemes are used. Of course, with the use of the CBS algorithm, an addition of a shock capturing viscosity is needed and many suitable schemes are discussed in [36].

Figure 4 shows the mesh used and the Mach number contours obtained for supersonic inviscid flow past a full cylinder. Here the inlet Mach number is 3. The Mach contour in Figure 4(b) is the result obtained from the MUSCL scheme [5]. Figure 4(c) and (d) are obtained using the CBS and different shock capturing viscosities. Using the appropriate shock capturing viscosity along with the CBS algorithm gives an excellent comparison with the special schemes [5]. Figure 5 shows the quantitative comparison of the present predictions with those of the special schemes [5]. It is seen that the present scheme agrees well with the special schemes except behind the cylinder, where all the schemes differ from each other slightly.

Figure 6 shows the mesh generated and the results obtained for hypersonic inviscid flow past a half cylinder. The inlet Mach number here is 10. The sensitive zone with recirculation behind the cylinder is well captured by this algorithm.

**Examples 5 and 6**

*Viscous compressible flow past a plate ( $M = 3.0$ ,  $Re = 1000$ ) and an NACA 0012 aerofoil ( $M = 0.95$ ,  $Re = 5000$ ).*

In both examples fully explicit schemes are used to solve supersonic viscous flow problems. In the first example (the Carter problem), the Mach number at the inflow is 3.0, the inlet Reynolds number based on the length of the plate is 1000. The temperature of the plate is assumed to be constant and equal to the stagnation temperature given as

$$T_s = T_\infty \left( 1 + \frac{\gamma - 1}{2} M_\infty^2 \right). \quad (54)$$

The temperature dependence of viscosity is accounted through Sutherland's law

$$\frac{\mu}{\mu_r} = \frac{T_r + S_0}{T + S_0} \left( \frac{T}{T_r} \right)^{1.5}, \quad (55)$$

where  $S_0$  is Sutherland's constant and is equal to 198.6° Rankine.

A uniform mesh with 13045 nodes and 6680 elements is used in this computation. The results obtained are shown in Figures 7 and 8. It is seen that the contours are smooth and shocks are in the proper location. A little underprediction in the pressure distribution (Figure 8(a)) can be attributed to the unstructured uniform mesh used in the study. However, the present results appear to be free from any oscillations, unlike Carter [37]. The outlet velocity profile agrees excellently (Figure 8(b)) with the Carter [37] results except near the shock.

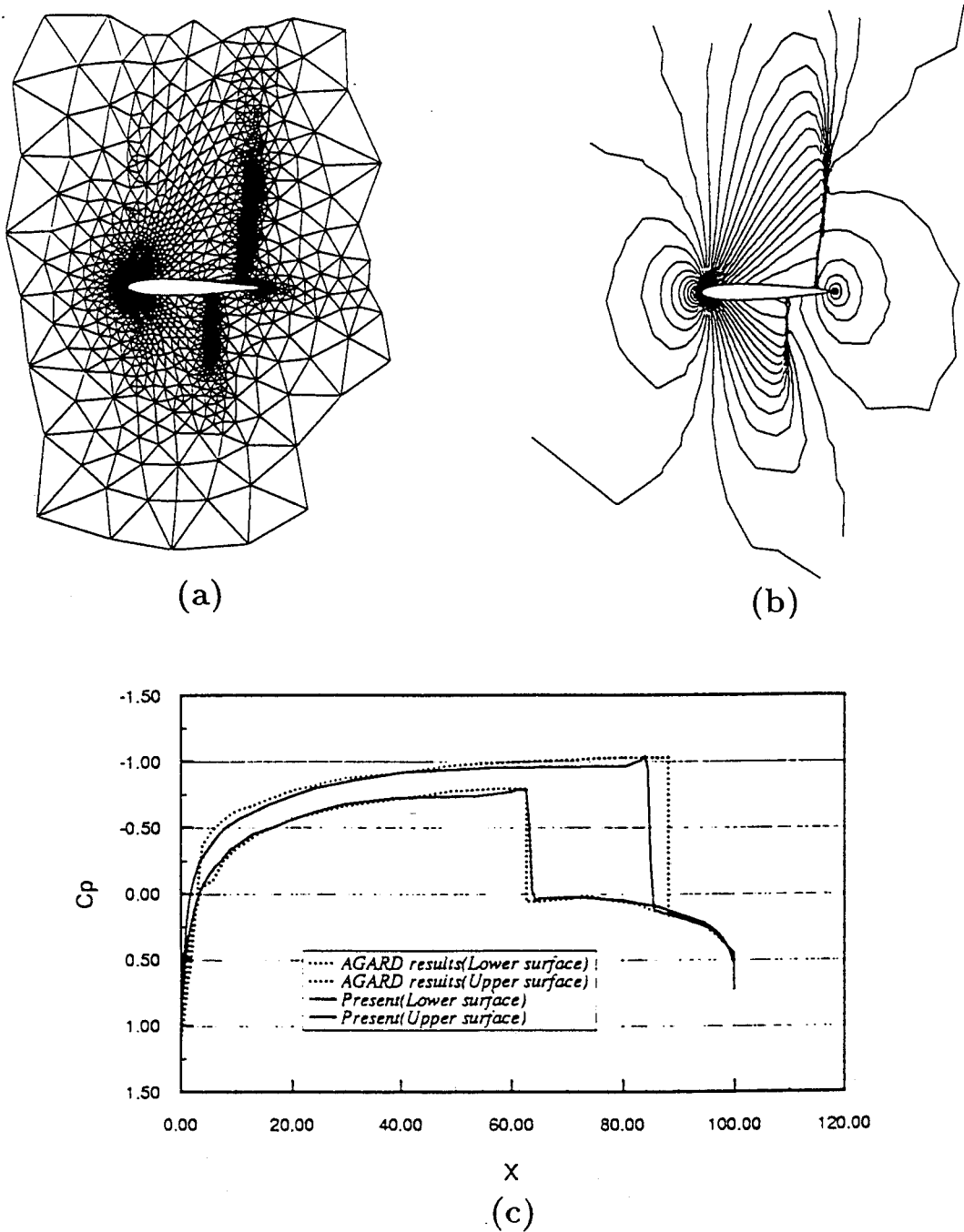


Figure 3. Inviscid flow past an NACA 0012 aerofoil with an angle of attack,  $\alpha = 1.2^\circ$ ,  $M = 0.85$ : (a) adapted mesh, 5512 elements and 2821 nodes; (b) pressure contours; (c) coefficient of pressure distribution along the upper and lower surfaces.



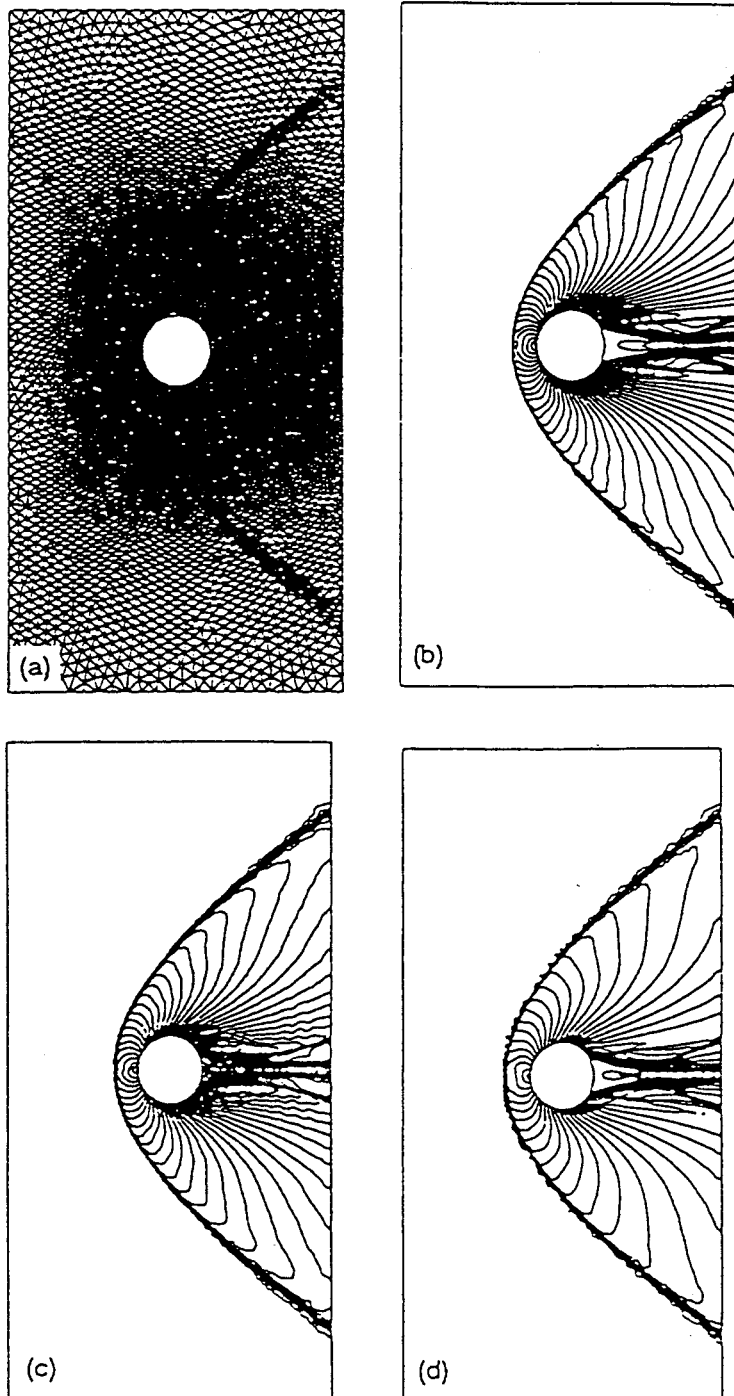


Figure 4. Supersonic inviscid flow past a full cylinder,  $\alpha = 0^\circ$ ,  $M = 3.0$ . (a) Adapted mesh, 24979 elements and 12651 nodes; (b) Mach contours with MUSCL scheme; (c) Mach contours with CBS scheme ( $M$  and  $M_L$  shock capturing); (d) Mach contours with CBS scheme (anisotropic shock capturing).

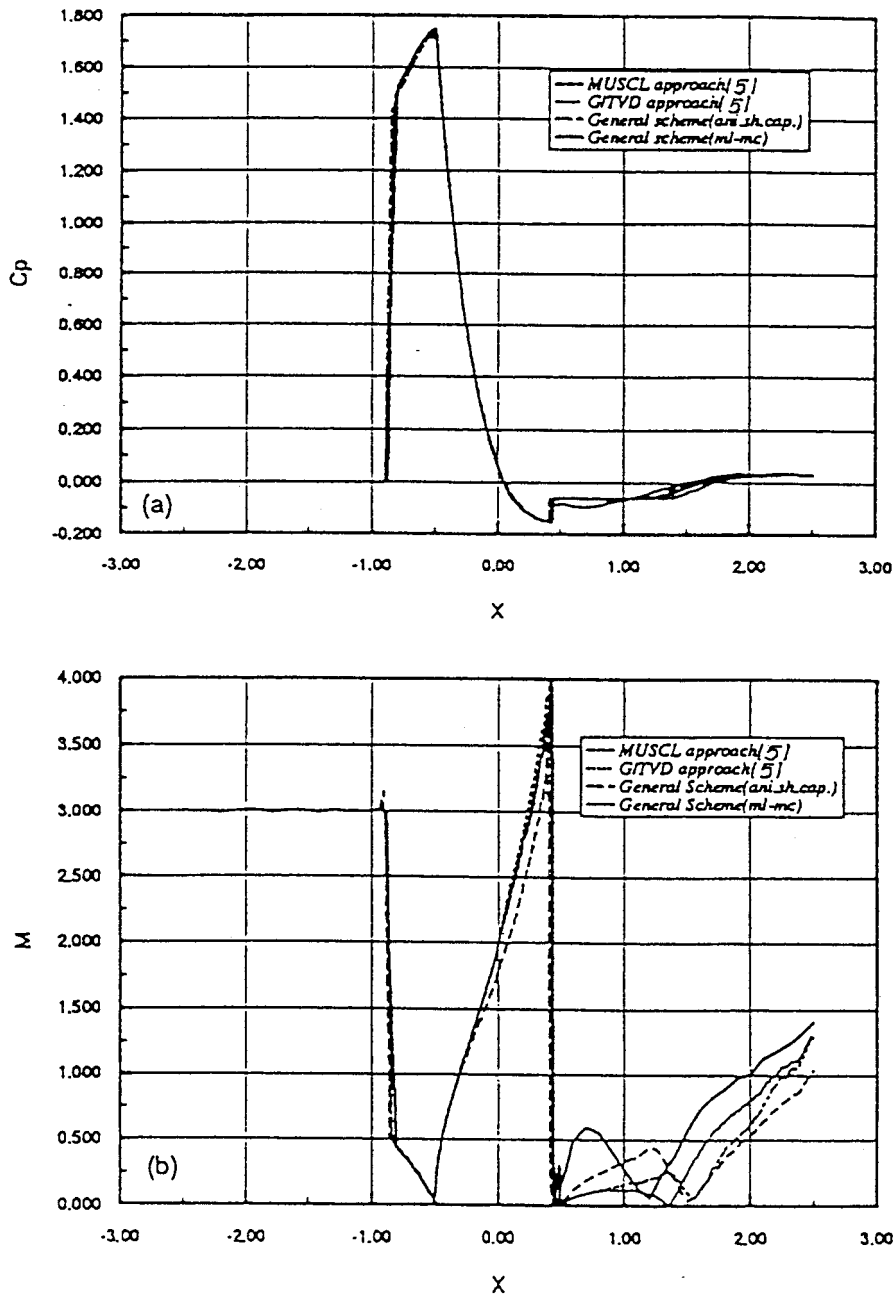
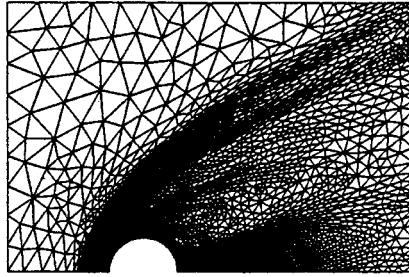
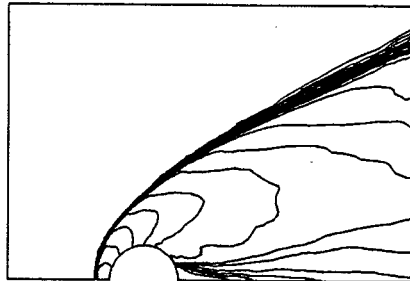


Figure 5. Comparison of coefficient of pressure and Mach number along the middle in horizontal direction of the domain.

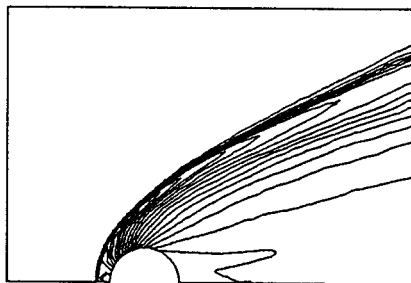
In the second problem, an adaptively graded mesh is used to resolve the wake region of the viscous, transonic flow past an NACA 0012 aerofoil. The total domain diameter is 25 times the chord of the aerofoil considered. The inlet Reynolds number is equal to 5000. The adaptive procedures are performed based on References [38,39]. Figure 9 and Plate 1 shows respectively



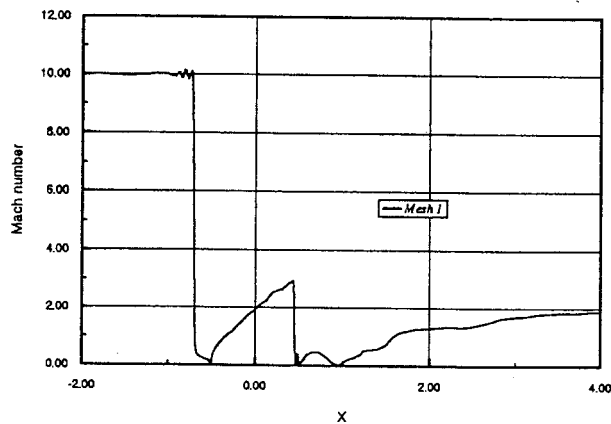
Mesh 1,  $h_{\min} = 0.035$ ,  $h_{\max} = 0.5$ ,  $C = 0.01$



Mach contours

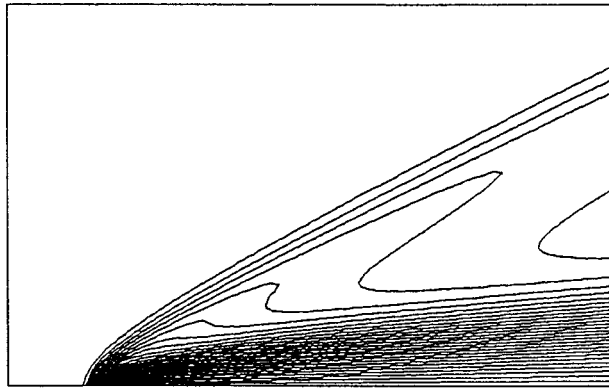


Density contours

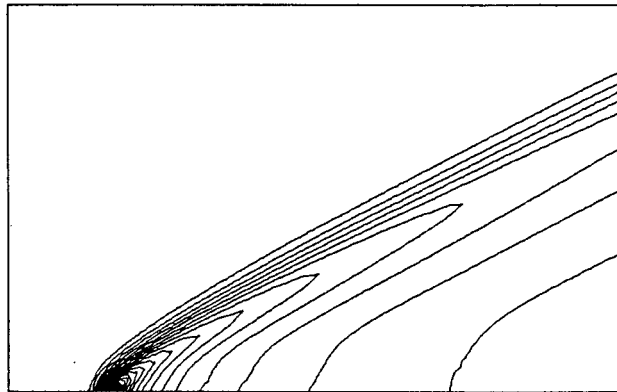


Mach number along bottom line

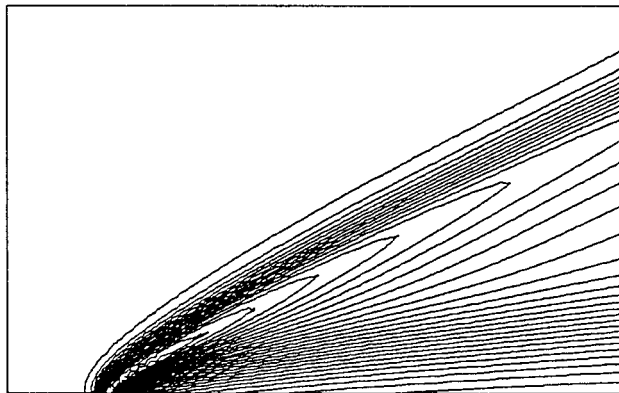
Figure 6. Hypersonic inviscid flow past a half cylinder,  $\alpha = 0^\circ$ ,  $M = 10.0$ . Adapted mesh and solution obtained.



(a)

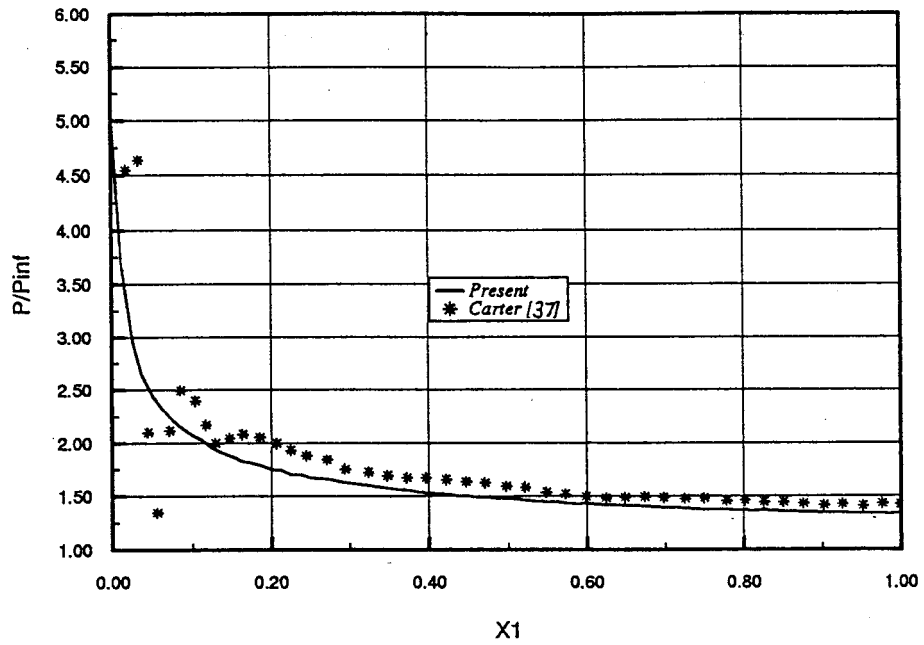


(b)

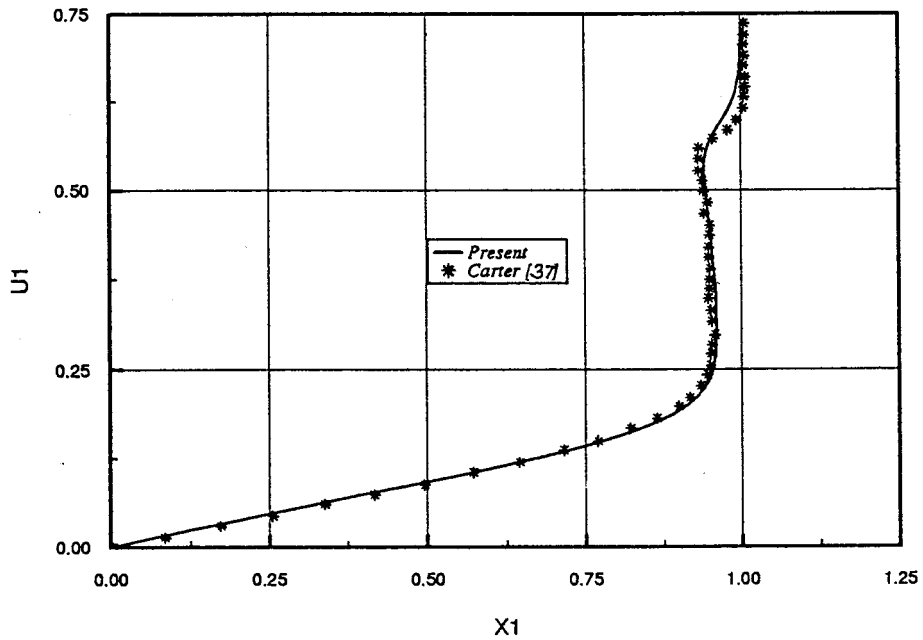


(c)

Figure 7. Supersonic viscous flow past a plate  $\alpha = 0^\circ$ ,  $M = 3.0$ ,  $Re = 1000$ : (a) Mach contours; (b) pressure contours; (c) density contours.



(a)



(b)

Figure 8. Comparison of pressure distribution and outlet velocity profile with Carter [37]. (a) Pressure distribution along the surface, (b) outlet velocity profile.

the mesh used and the density contours. It is seen that the instability predicted in this example well matches the one predicted using more complicated adaptive procedures [40].

### 6.2. Semi-implicit procedure

Below, three examples of a semi-implicit nature are presented.

#### Example 1

*High Reynolds number flow in a lid-driven cavity,  $M = 0$ .*

The first example is a fully viscous incompressible flow ( $M = 0$ ) in a lid-driven cavity. This is a well-known test case used by many authors. A high-Reynolds number of 5000 is used in this study. An adapted mesh taken from [41] is used to study the problem. The number of nodes and elements used are much lower than the one used in the normal computations [4,6].

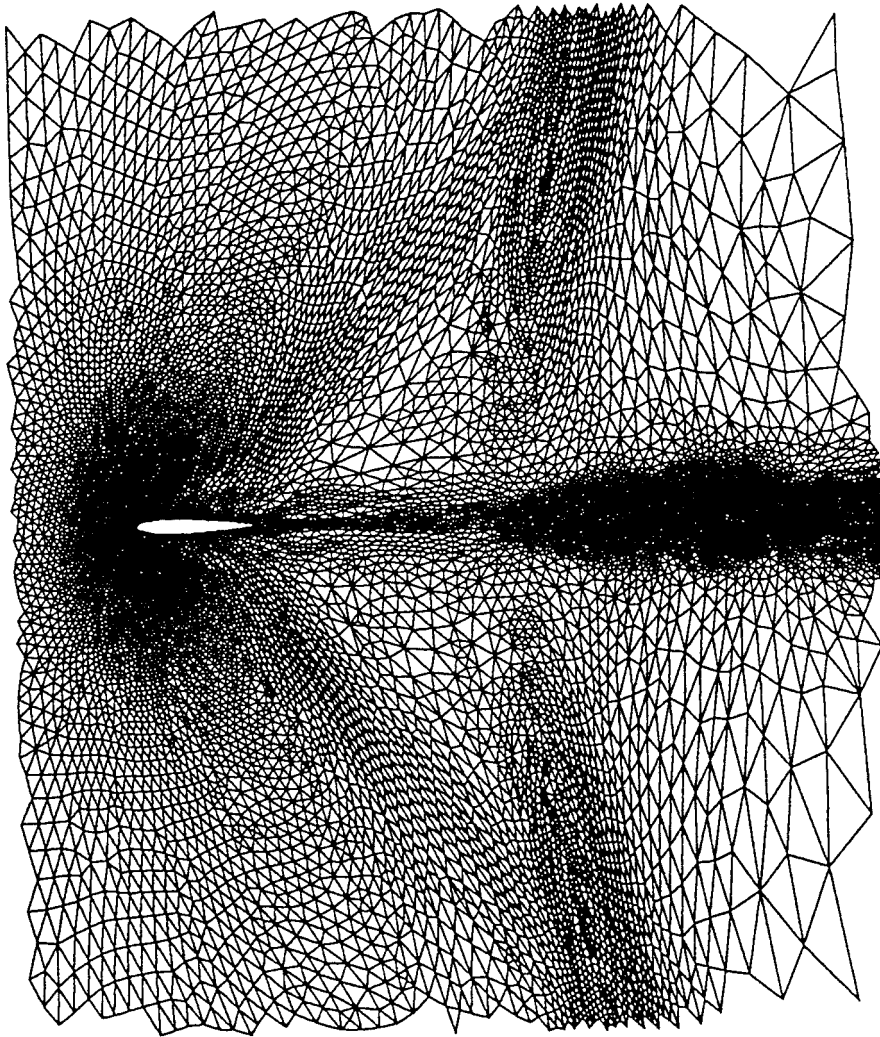
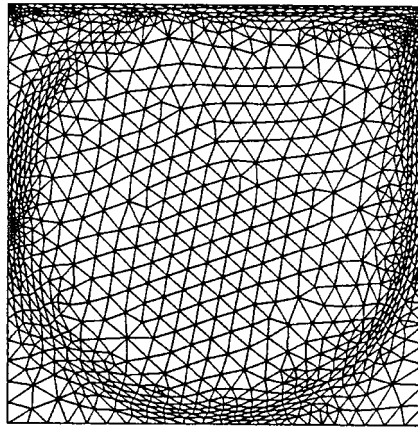
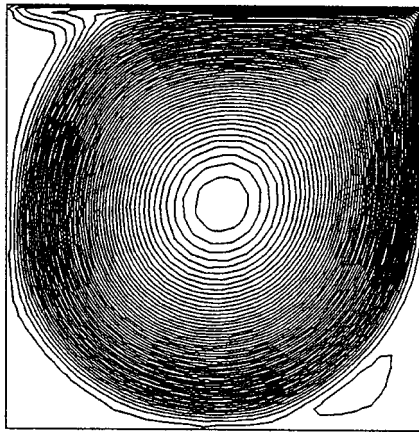


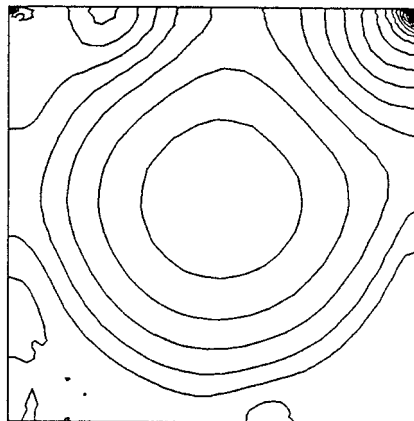
Figure 9. Adapted mesh for transonic, viscous flow past an NACA 0012 aerofoil  $\alpha = 0^\circ$ ,  $M = 0.95$ ,  $Re = 5000$  (elements = 32522, nodes = 16388).



(a)



(b)



(c)

Figure 10. Viscous incompressible flow in a lid-driven cavity,  $Re = 5000$ . (a) Adapted mesh, elements = 1962, nodes = 1034; (b) streamlines; (c) pressure contours.

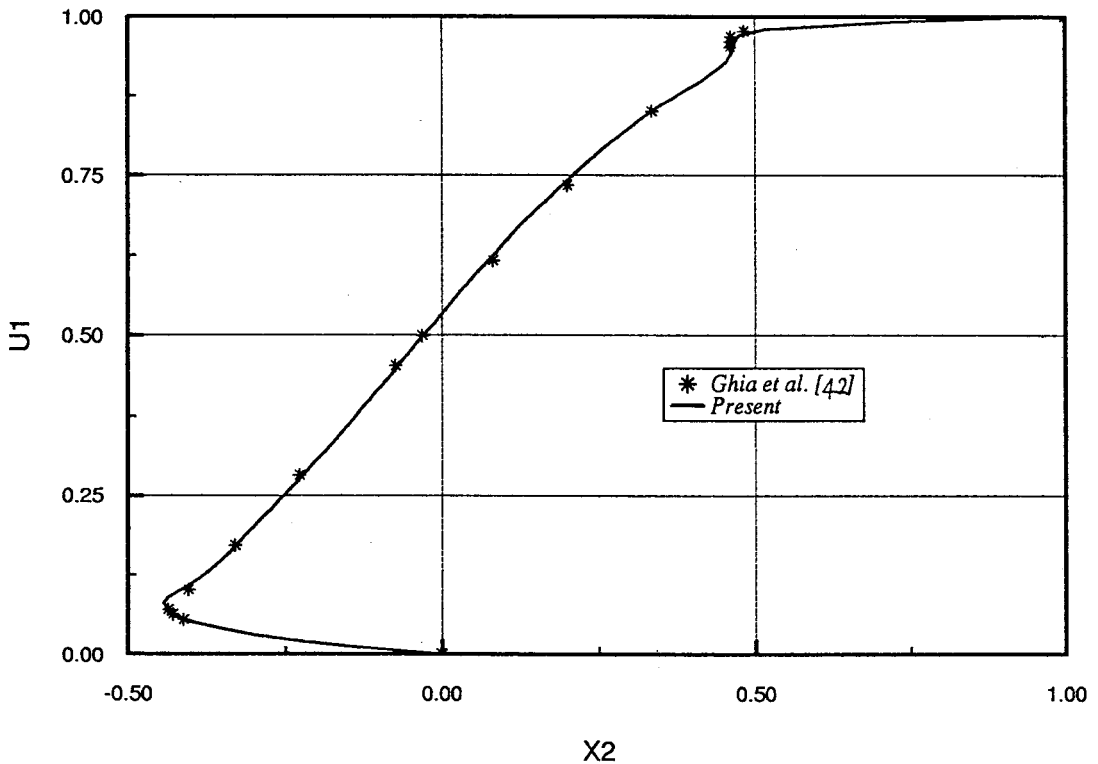


Figure 11. Comparison of present velocity distribution with the benchmark solution of Ghia *et al.* [42].

The mesh used and the streamline and pressure contours are shown in Figure 10. The recirculation near the corners are well predicted by the scheme and compares well with the benchmark solution of Giha *et al.* [42]. The velocity distribution across the cavity is compared with the benchmark solution in Figure 11. The agreement is excellent.

### Example 2

*Incompressible flow past a backward-facing step,  $M = 0$ .*

This is another benchmark problem for which experimental results are available. Here also, the adapted mesh is used to solve the flow. The inlet velocity profile is assumed to be parabolic and the inlet Reynolds number is equal to 229. The total length of the domain is taken equal to 40 times the step height so that traction-free conditions are valid at the outlet.

Figure 12 shows the mesh used and the solution obtained. The pressure distribution is smooth and the comparison of velocity profiles with the experiments [43] is good. The small deviation from the experimental prediction can be attributed to the inlet parabolic profile used in the present study. The referred experimental results are generated from an inlet profile slightly different from the parabolic profile.

### Example 3

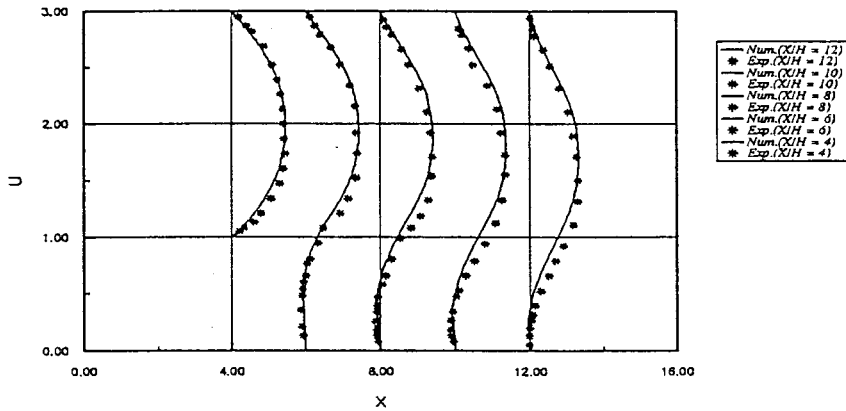
*Inviscid transonic compressible flow past an NACA 0012 aerofoil,  $M = 0.5$ .*

In this example, the inlet Mach number is assumed to be equal to 0.5. The velocity is prescribed at the inlet and the density is prescribed at the exit. The density is calculated analytically using the following relation





(a) Mesh  $h_{\min} = 0.05$ ,  $h_{\max} = 0.5$ ,  $C = 0.15$ ,  $NNODES = 1820$ ,  $NELEM = 3441$



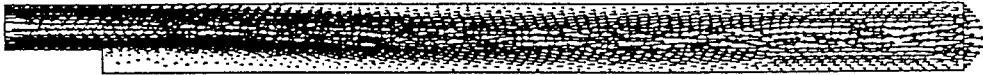
(b) Comparison of velocity profiles



(c) Stream line contours



(d) Pressure contours



(e) Velocity vector plot

Figure 12. Incompressible viscous flow past a backward-facing step,  $Re = 229$ , finite element mesh and solution obtained. (a) Mesh  $h_{\min} = 0.05$ ,  $h_{\max} = 0.5$ ,  $C = 0.15$ , number of nodes = 1820, number of elements = 3441; (b) comparison of velocity profiles; (c) streamline contours; (d) pressure contours; (e) velocity vector plot.

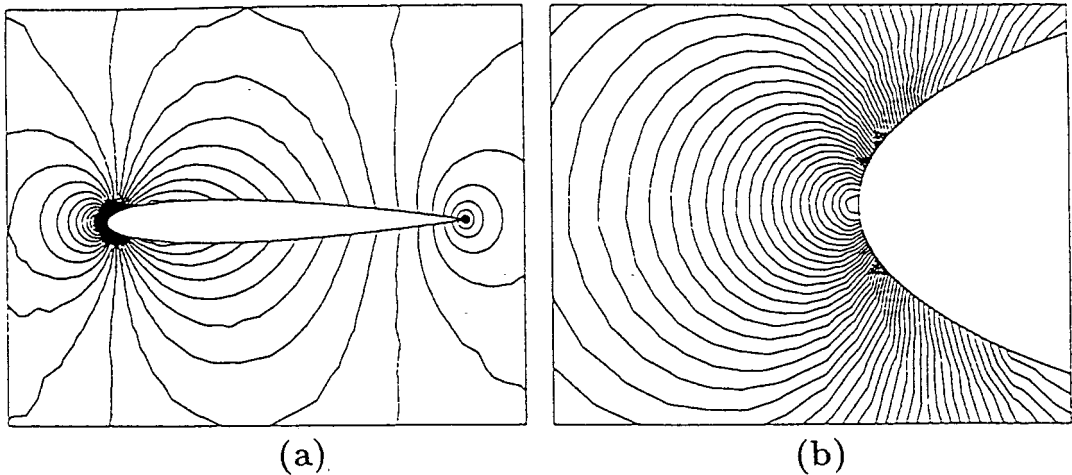


Figure 13. Subsonic inviscid flow past an NACA 0012 aerofoil, semi-implicit solution,  $\alpha = 0^\circ$ ,  $M = 0.5$ . (a) Pressure contours; (b) pressure distribution near stagnation point.

$$\rho_0 = \rho_\infty \left( 1 + \frac{\gamma - 1}{2} M_\infty^2 \right)^{1/(\gamma - 1)}, \quad (56)$$

which gives a value of 1.1297 and the present computation gives 1.1320 and the difference is negligibly small.

Figure 13 shows the pressure contours predicted by the present scheme. The oscillation-free pressure distribution near the stagnation point shows the excellent performance of the CBS algorithm in its semi-implicit form.

### 6.3. Implicit procedure

An example of an implicit nature is presented.

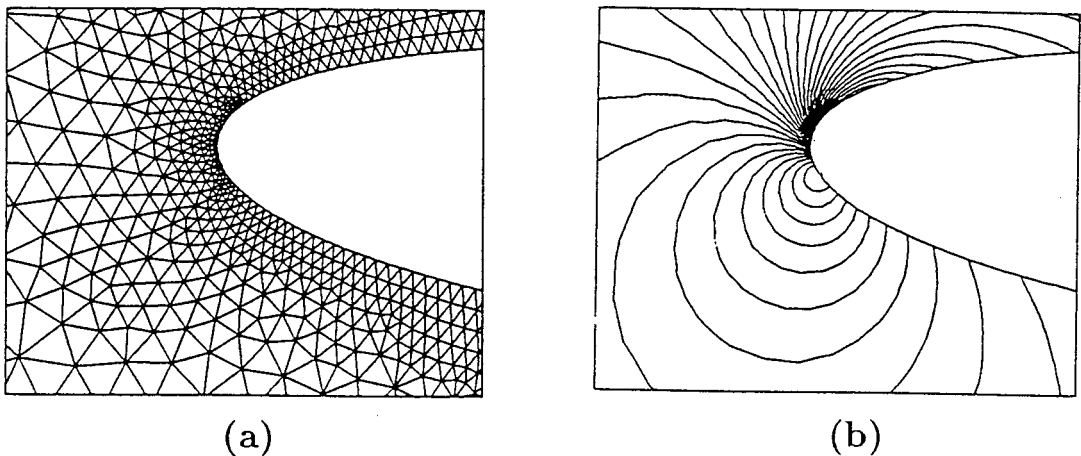
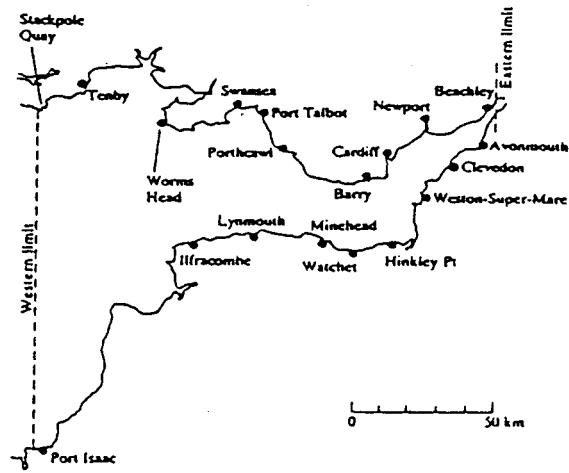
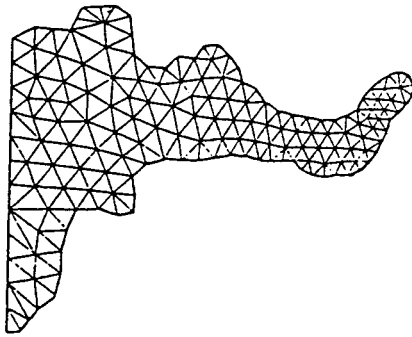


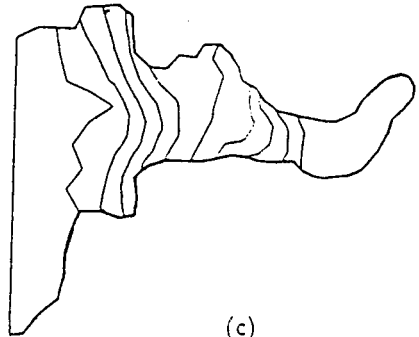
Figure 14. Incompressible inviscid flow past an NACA 0012 aerofoil, implicit solution,  $\alpha = 5^\circ$ ,  $M = 0$ . (a) Mesh near stagnation point; (b) pressure contours.



(a)



(b)



(c)

Figure 15. The Severn Estuary. (a) Analysis domain; (b) finite element mesh; (c) bathymetry.

### Example 1

*Inviscid incompressible flow past an NACA 0012 aerofoil with an angle of attack,  $M = 0$ .*

In this example a fully implicit form of the solution procedure is used to solve the inviscid flow past an NACA 0012 aerofoil with an angle of attack equal to  $5^\circ$ . Here, although an additional system of equations needs to be solved per time step, the time step may be taken much larger, making the total CPU time needed much smaller. The mesh used near the stagnation point and the pressure contours obtained are shown in Figure 14. The comparison between the analytical stagnation and the present prediction is very close and the deviation is only about 3%. Further details of the implicit procedure can be found elsewhere [7].

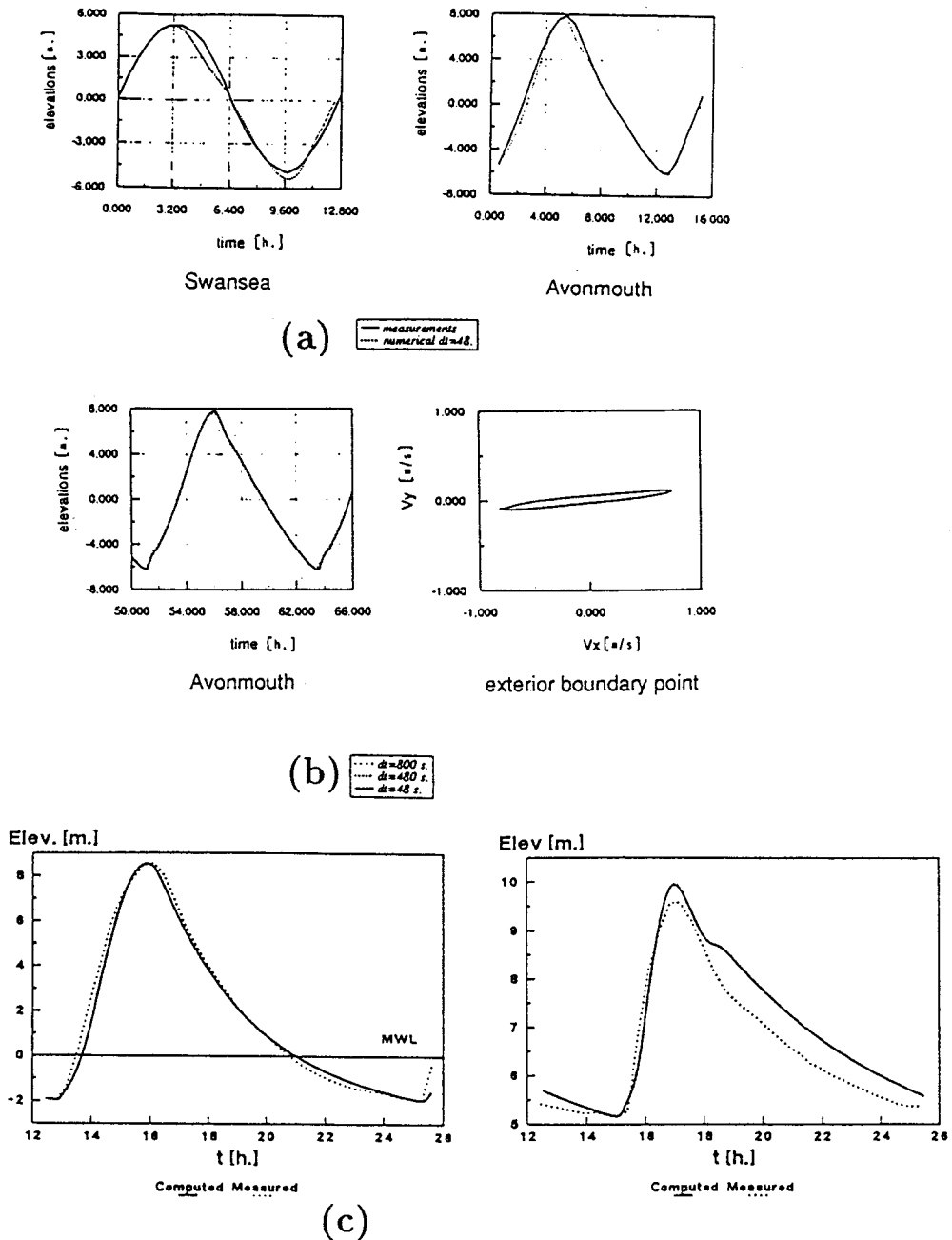


Figure 16. (a) Comparison of computed and measured surface elevation; (b) comparison of computed results for explicit and semi-implicit schemes; (c) comparison of computed and measured elevations at two points in Severn Bore.

#### 6.4. Shallow water problems

We have mentioned that for this class of problem, the first efficient use of the fully explicit form was made by Peraire *et al.* [22]. For tidal problems, however, the semi-implicit form is highly desirable and much larger time steps can be used.

Table I. Comparison of present results with experiments

Station	Observations (HRS)			Model (d.t. = 50 s)	
	$a$ (cm)	$g$ (d)	$a$ (cm)	$g$ (d)	% ( $a$ )
Boundary (mid)	235	160	238	160 (0)	1.2
Tenby	262	170	254	170 (0)	3
Swansea	315	173	307	176 (+3)	2.5
Ilfracombe	308	162	300	165 (+5)	2.6
Barry	382	182	387	186 (+4)	1.3
Porthcawl	317	173	327	178 (+5)	3
Avonmouth	422	202	432	191 (−11)	2.3

In Figures 15 and 16 we show an analysis of tidal flows in the Severn Estuary and both results of the explicit and semi-implicit procedures are given. Table I shows the comparison of the predicted and experimental values.

Plate 2 shows a steady state supercritical flow in an open channel in colour. Here, just like supersonic flows, a fully explicit scheme is desirable.

### 6.5. Solid dynamic problems

We conclude this section of examples by a somewhat unusual elesto-plastic solid and the CBS scheme is used entirely to make use of its capabilities of overcoming the BB-type restrictions. Such restrictions allowed so far only to use rectangles and hexahedra and a single point integration to be used as triangles and tetrahedra were not effective.

The present scheme allows both triangles and tetrahedra to be used in such codes as DYNA 3D, as Figures 17 and 18 show the use of such elements in a dynamic problem.

## 7. CONCLUDING REMARKS

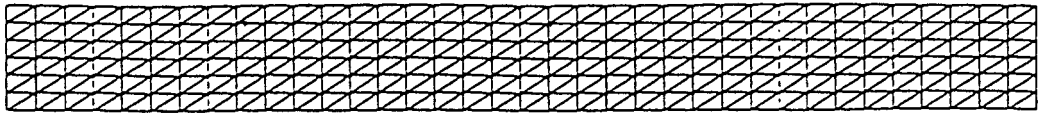
We hope that in this paper we have introduced the logical background and the excellent performance of the CBS algorithm.

The development of the algorithm has taken some time and the present authors would like to acknowledge the earlier works which pointed the way to the final algorithm. In our original paper [1], we discuss the SEARCH process and mentioned the early papers that were only partly successful. However, we borrow from the title of the first of these early ventures a title that well describes the present procedure as

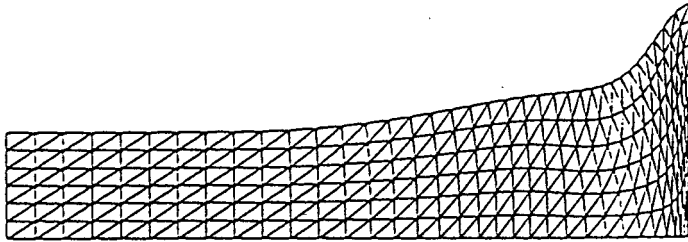
*‘AN ALGORITHM FOR ALL SEASONS’*

### ACKNOWLEDGMENTS

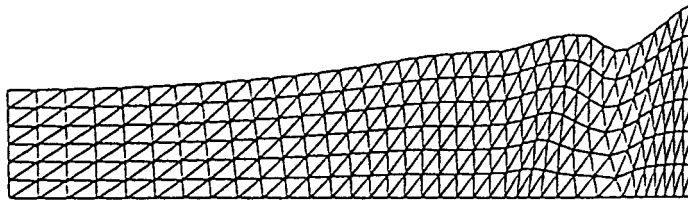
This research has been partially supported by NASA grant NAGW/2127, AMES Control Number 90-144.



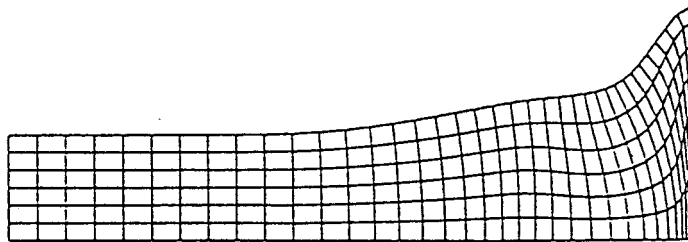
a) Initial shape with the finite element discretization



b) Linear triangles — CBS Algorithm



c) Linear triangles — Displacement Algorithm



d) Bilinear quadrilaterals — DYNA Algorithm

Figure 17. Impact of plastic bar—axisymmetric solution. (a) Initial shape with the finite element discretization; (b) linear triangles—CBS algorithm; (c) linear triangles—displacement algorithm; (d) bilinear Quadrilaterals—DYNA algorithm.

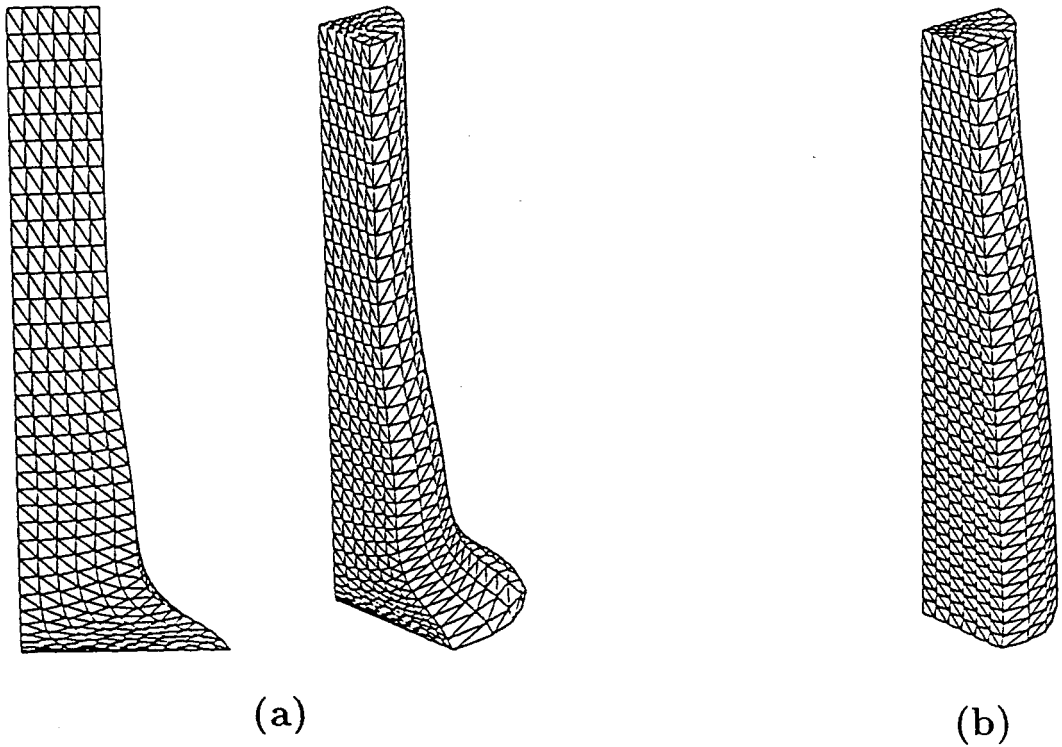


Figure 18. Impact of a plastic bar—three-dimensional solution. (a) CBS algorithm; (b) standard displacement algorithm.

## APPENDIX A. SHALLOW WATER DEPTH-AVERAGED EQUATIONS

These equations reduce the problem to a two-dimensional one considering only averaged velocities in the vertical direction.

If  $x_1$  and  $x_2$  are the horizontal co-ordinates, Figure A1 defines the basic variables.

The equations are identical to those of a two-dimensional compressible flow without the energy equation. Thus, we can write

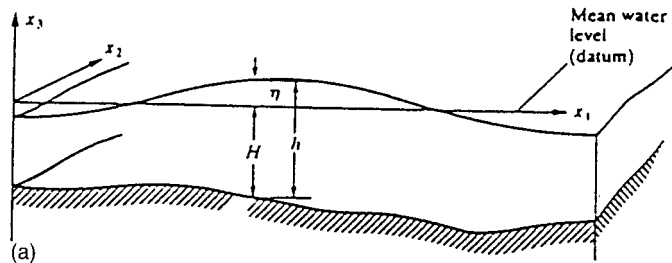


Figure A1. Shallow water formulation.

$$\begin{aligned} \frac{1}{c^2} \frac{\partial p}{\partial t} + \frac{\partial U_i}{\partial x_i} &= 0, \\ \frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j} (u_j U_k) + \frac{\partial p}{\partial x_i} + Q_i &= 0 \quad (i, j = 1, 2), \end{aligned} \quad (57)$$

where

$$U_i = hu_i \quad (58)$$

denotes the mass flow,

$$c^2 = gh \quad (59)$$

is the wave speed, and

$$p = \frac{1}{2} g(h^2 - H^2) \quad (60)$$

corresponds to the pressure, with  $h$ ,  $H$  and  $g$  defined in Figure A1.

The source term is defined below

$$Q_i = -g(h - H) \frac{\partial H}{\partial x_i} + g \frac{u_i |u_i|}{C^2 h} + r_i - \tau_i + \frac{h}{\rho} \frac{\partial p_c}{\partial x_i}. \quad (61)$$

Here, the bed friction force is defined by the Chezy formula (with  $C$  being the Chezy constant,  $r$  is the Coriolis force), with the vector  $r$  is defined as

$$r = \begin{bmatrix} -fu_2 \\ fu_1 \end{bmatrix}, \quad (62)$$

with  $f$  being the Coriolis friction. Finally,  $\tau_i$  is the wind surface traction.

Once again sub and supersonic phenomena can occur as shown in the examples.

## APPENDIX B. BOUNDARY CONDITIONS

In most of the problems we generally have ‘open’ boundaries limiting the extent of the analysis domain. On such boundaries, various approximations are conventionally used. It is not our purpose to discuss this type of boundary but the reader should refer to procedures used conventionally used in the downstream region [8,44].

Our interest, however, is the boundary conditions on solid walls and we discuss these below.

### B.1. Boundary conditions on solid walls

(a) On solid boundaries

$$u_n = 0 \quad \text{normal velocity zero} \quad (63)$$

and either

$$t_s = 0 \quad \text{tangential traction zero}$$

or

$$u_s = 0 \quad \text{tangential velocity zero} \quad (64)$$



(b) When computing  $\Delta \bar{U}^*$  we do not impose any part at  $\Delta \bar{U}^* = 0$ , but leave values to be determined. However, traction conditions on deviatoric components must now be applied to this and appropriate values of  $t_s$  and  $t_n$  must be calculated on the whole boundary. This requires, in viscous problems, the evaluation of  $\tau_{ij}$ .

(c) When computing  $\Delta p$ , we integrate by parts obtaining

$$\begin{aligned} \int_{\Omega} N^T \Delta p \, d\Omega &= -\Delta t \int_{\Omega} N^T \frac{\partial}{\partial x_i} \left( \bar{U}_i^n + \theta_1 \Delta \bar{U}_i - \theta_1 \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \right) d\Omega \\ &= -\Delta t \int_{\Omega} N^T \frac{\partial}{\partial x_i} \bar{U}_i^n + \Delta t \theta_1 \int_{\Omega} \frac{\partial N^T}{\partial x_i} \left( \Delta \bar{U}_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \right) d\Omega \\ &\quad - \Delta t \theta_1 \int_{\Gamma} N^T n_i \left( \Delta \bar{U}_i^* - \Delta t \frac{\partial p^{n+\theta_2}}{\partial x_i} \right). \end{aligned} \quad (65)$$

Here,  $n_i$  is the outward drawn normal. The last term in the above equation is equal to zero, as is the condition of Equation (63), Stokes flow

$$u_n = n_i \Delta \bar{U}_i = n_i \left( \Delta \bar{U}_i^* - \Delta t \frac{\partial p}{\partial x_i} \right) = 0 \quad (66)$$

to the order of  $\Delta t^2$ . This point seems to have baffled some investigators who simply assume

$$\frac{\partial p}{\partial x_i} = 0 \quad (67)$$

on solid boundaries. Note that this is not exactly true.

## REFERENCES

1. O.C. Zienkiewicz and R. Codina, 'Search for a general fluid mechanics algorithm', in D.A. Caughey and M.M. Hafez (eds.), *Frontiers of Computational Fluid Dynamics*, Wiley, New York, 1995, pp. 101–113.
2. O.C. Zienkiewicz and R. Codina, 'A general algorithm for compressible and incompressible flow, part I. The split characteristic based scheme', *Int. J. Numer. Methods Fluids*, **20**, 869–885 (1995).
3. O.C. Zienkiewicz, B.V.K. Satya Sai, K. Morgan, R. Codina and M. Vázquez, 'A general algorithm for compressible and incompressible flow part II. Tests on the explicit form', *Int. J. Numer. Methods Fluids*, **20**, 887–913 (1995).
4. R. Codina, M. Vázquez and O.C. Zienkiewicz, 'General algorithm for compressible and incompressible flows, part III. A semi-implicit form', *Int. J. Numer. Methods Fluids*, **27**, 13–32 (1998).
5. B.V.K. Satya Sai, O.C. Zienkiewicz, M.T. Manzari, P.R.M. Lyra, and K. Morgan, 'General purpose vs. special algorithms for high speed flows with shocks', *Int. J. Numer. Methods Fluids*, **27**, 57–80 (1998).
6. O.C. Zienkiewicz, B.V.K. Satya Sai, K. Morgan and R. Codina, 'Split characteristic based semi-implicit algorithm for laminar/turbulent incompressible flows', *Int. J. Numer. Methods Fluids*, **23**, 1–23 (1996).
7. R. Codina, M. Vázquez and O.C. Zienkiewicz, 'A fractional step method for the solution of compressible Navier–Stokes equations', in M. Hafez and K. Oshima (eds.), *Computational Fluid Dynamics Review 1998*, Vol. 1, World Scientific, Singapore, 1998, pp. 331–347.
8. O.C. Zienkiewicz, J. Szmelter and J. Peraire, 'Compressible and incompressible flow: an algorithm for all seasons', *Comput. Method Appl. Mech. Eng.*, **78**, 105–121 (1990).
9. O.C. Zienkiewicz, 'Explicit or semi-explicit general algorithm for compressible and incompressible flows with equal finite element interpolation', *Report 90/5*, Chalmers University of Technology, 1990.
10. O.C. Zienkiewicz, 'Finite elements and computational fluid mechanics', *Metodos Numericos en Ingenieria*, SEMNI Congress, Gran Canaria, 1991, pp. 56–61.
11. O.C. Zienkiewicz and J. Wu, 'Incompressibility without tears! How to avoid restrictions of mixed formulation', *Int. J. Numer. Methods Eng.*, **32**, 1184–1203 (1991).
12. O.C. Zienkiewicz and J. Wu, 'A general explicit of semi-explicit algorithm for compressible and incompressible flows', *Int. J. Numer. Methods Eng.*, **35**, 457–479 (1992).
13. P.D. Lax and B. Wendroff, 'Systems of conservation laws', *Comm. Pure. Appl. Math.*, **13**, 217–237 (1960).

14. J. Donea, S. Giuliani, H. Laval and L. Quartapelle, 'Time-accurate solution of advection-diffusion problems by finite elements', *Comput. Methods Appl. Mech. Eng.*, **45**, 123-145 (1984).
15. J. Donea, 'A Taylor-Galerkin method for convective transport problems', *Int. J. Numer. Methods Eng.*, **20**, 101-119 (1984).
16. R. Löhner, K. Morgan and O.C. Zienkiewicz, 'The solution of non-linear hyperbolic equation systems by the finite element method', *Int. J. Numer. Methods Fluids*, **4**, 1043-1063 (1984).
17. O.C. Zienkiewicz, R. Löhner, K. Morgan and J. Peraire, 'High speed compressible flow and other advection dominated problems of fluid mechanics', in R.H. Gallagher, G.F. Carey, J.T. Oden and O.C. Zienkiewicz (eds.), *Finite Elements in Fluids*, vol. 6, Chap. 2, Wiley, New York, 1985, pp. 41-88.
18. A.J. Chorin, 'Numerical solution of Navier-Stokes equations', *Math. Comput.*, **22**, 745-762 (1968).
19. A.J. Chorin, 'On the convergence of discrete approximation to the Navier-Stokes equations', *Math. Comput.*, **23**, 341-353 (1969).
20. G.L. Goudreau and J.O. Hallquist, 'Recent developments in large scale Lagrangian hydrocodes', *Comput. Methods Appl. Mech. Eng.*, **33**, 725-757 (1982).
21. O.C. Zienkiewicz, J. Rojek, R.L. Taylor and M. Pastor, 'Triangles and tetrahedra in explicit dynamic codes for solids', *Int. J. Numer. Methods Eng.*, **43**, 565-583 (1998).
22. J. Peraire, O.C. Zienkiewicz and K. Morgan, 'Shallow water problems: a general explicit formulation', *Int. J. Numer. Methods Eng.*, **22**, 547-574 (1986).
23. O.C. Zienkiewicz and P. Ortiz, 'A split-characteristic based finite element model for shallow water equations', *Int. J. Numer. Methods Fluids*, **20**, 1061-1080 (1995).
24. O.C. Zienkiewicz and P. Ortiz, 'An improved finite element model for shallow water problems', in G.F. Carey (ed.), *Finite Element Modelling of Environmental Problems*, Wiley, New York, 1996, pp. 61-84.
25. P. Ortiz and O.C. Zienkiewicz, 'Tide and bore propagation in the Severn Estuary by a new fluid algorithm', *Proc. Int. Conf. on Finite Elements Fluids—New Trends and Applications*, Venezia, 1995, pp. 1543-1552.
26. R.A. Adey and C.A. Brebbia, 'Finite element solution of effluent dispersion', in C.A. Brebbia and J.J. Connor (eds.), *Numerical Methods in Fluid Mechanics*, Pentech Press, 1974, pp. 325-354.
27. R.E. Ewing and T.F. Russell, 'Multistep Galerkin methods along characteristics of convection-diffusion problems', in R. Vichnevetsky and R.S. Stepleman (eds.), *Advanced Computer Methods in PDEs IV*, IMACS, Rutgers University, 1981, pp. 28-36.
28. O. Pironneau, 'On the transport diffusion algorithm and its application to the Navier-Stokes equation', *Numer. Math.*, **38**, 309-332 (1982).
29. M. Bercovier, O. Pironneau, Y. Harbani and E. Livne, 'Characteristics and finite element methods applied to equations of fluids', in J.R. Whiteman (ed.), *The Mathematics of Finite Elements and Applications*, Vol. V, Academic Press, New York, 1982, pp. 471-478.
30. M. Bercovier, O. Pironneau and V. Sastri, 'Finite elements and characteristics for some parabolic-hyperbolic problems', *Appl. Math. Model.*, **7**, 89-96 (1983).
31. J.P. Benque, J.P. Gregoire, A. Hauguel and M. Maxant, 'Application des methodes du decomposition aux calculs numeriques an hydraulique industrielle', *INRIA 4th Coll. Methodes de Calcul Sci. et Techn.*, Versailles, 1983.
32. O.C. Zienkiewicz and R.L. Taylor, *The Finite Element Method*, Vol. 2, 4th edition, McGraw-Hill, New York, 1991.
33. P. Oñate, 'Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems', *Comput. Methods Appl. Mech. Eng.*, **151**, 233-265 (1998).
34. R. Codina, 'Stability analysis of forward Euler scheme for the convection-diffusion equation using the SUPG formulation in space', *Int. J. Numer. Methods Eng.*, **36**, 1445-1464 (1993).
35. T.H. Pulliam and J.T. Barton, 'Euler computations of AGARD working group 07 aerofoil test cases', AIAA 23rd Aerospace Sciences Meeting, Reno, NV, 14-17 January 1985.
36. P. Nithiarasu, O.C. Zienkiewicz, B.V.K. Satya Sai, K. Morgan, R. Codina and M. Vázquez, 'Shock capturing viscosities for the general fluid mechanics algorithm', *Int. J. Numer. Methods Fluids*, **28**, 1325-1353 (1998).
37. J.E. Carter, 'Numerical solutions of the Navier-Stokes equations for the supersonic laminar flow over a two-dimensional compression corner', *NASA TR-R-385*, 1972.
38. J. Peraire, M. Vahdati, K. Morgan and O.C. Zienkiewicz, 'Adaptive remeshing for compressible flow computations', *J. Comput. Phys.*, **72**, 449-466 (1987).
39. O.C. Zienkiewicz and J. Wu, 'Automatic directional refinement in adaptive analysis of compressible flows', *Int. J. Numer. Methods Eng.*, **37**, 2189-2210 (1994).
40. H. Borouchaki, P.L. George, F. Hecht, P. Laug and E. Saltel, 'Delaunay mesh generation governed by metric specifications. Part I. Algorithms', *Finite Elem. Anal. Des.*, **25**, 61-83 (1997).
41. P. Nithiarasu and O.C. Zienkiewicz, 'Towards better choices of adaptive mesh generation for fluid mechanics problems', *6th ACME-UK Conference*, 6-7 April 1998, pp. 129-132.
42. U. Ghia, K.N. Ghia and C.T. Shin, 'High-Re solution for incompressible flow using the Navier-Stokes equations and multigrid method', *J. Comput. Phys.*, **48**, 387-411 (1982).
43. M.K. Denham and M.A. Patrick, 'Laminar flow over a downstream facing step in a two-dimensional channel', *Trans. Inst. Chem. Eng.*, **52**, 361-367 (1974).
44. T.C. Papanastasiou, N. Malamataris and K. Ellwood, 'A new outflow boundary condition', *Int. J. Numer. Methods Fluids*, **14**, 587-608 (1992).

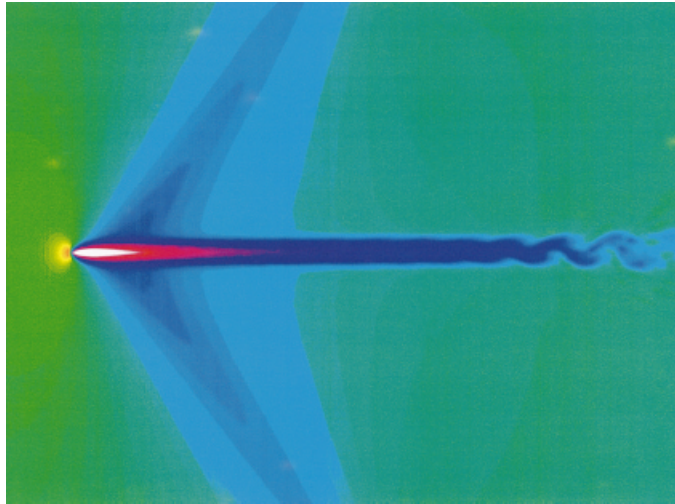


Plate 1. Density contours.

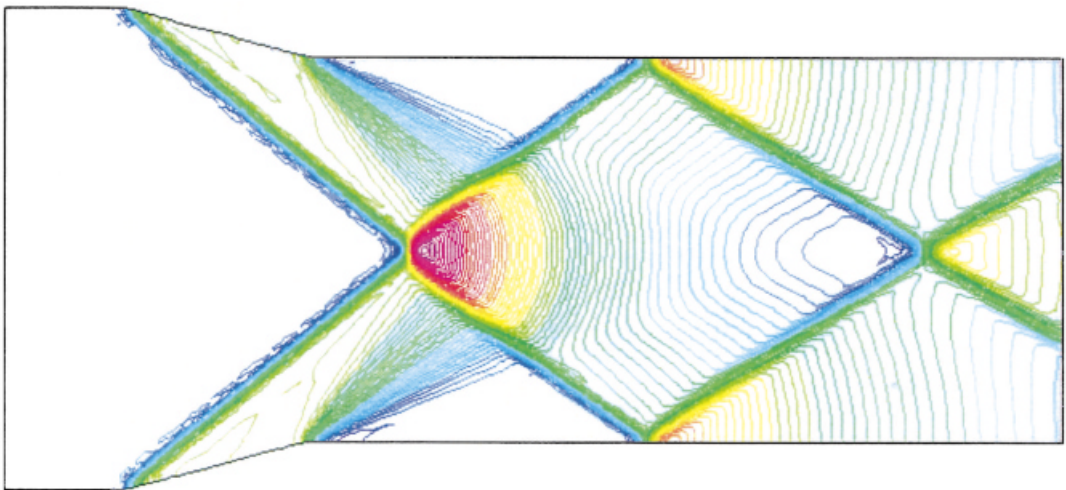


Plate 2. Symmetric channel of variable width, contours of  $h$ , Froude number = 2.5.